百度智能云

大模型时代的异构计算平台

资深研发工程师 孙鹏

演讲提纲

百度智能云

01 GPT-3开启大模型时代

02 超大模型训练对基础设施的需求

03 软硬结合的联合优化

04 大模型发展推动基础设施演进

百度智能云

01

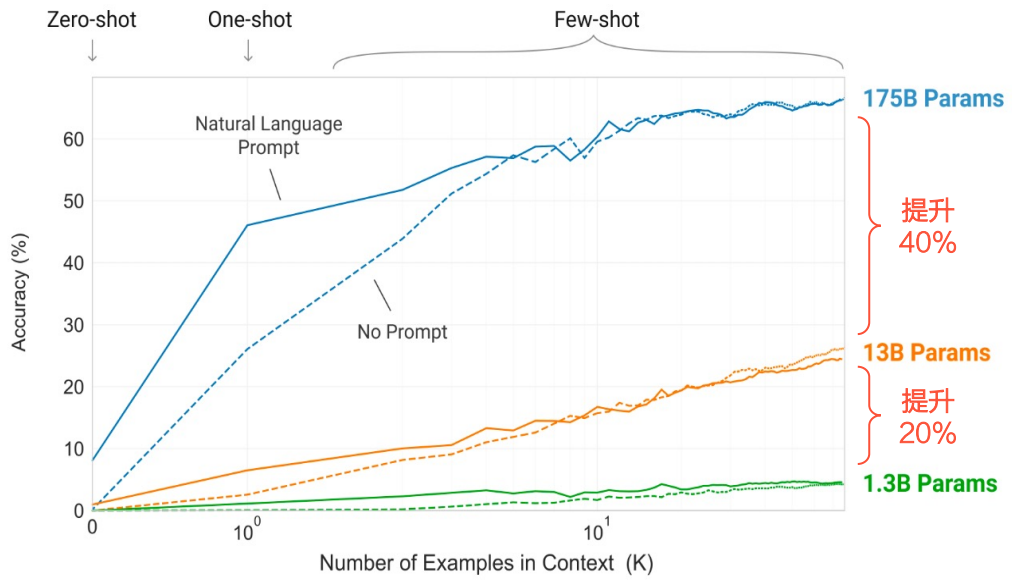
GPT-3开启大模型时代

大模型带来质的效果飞跃

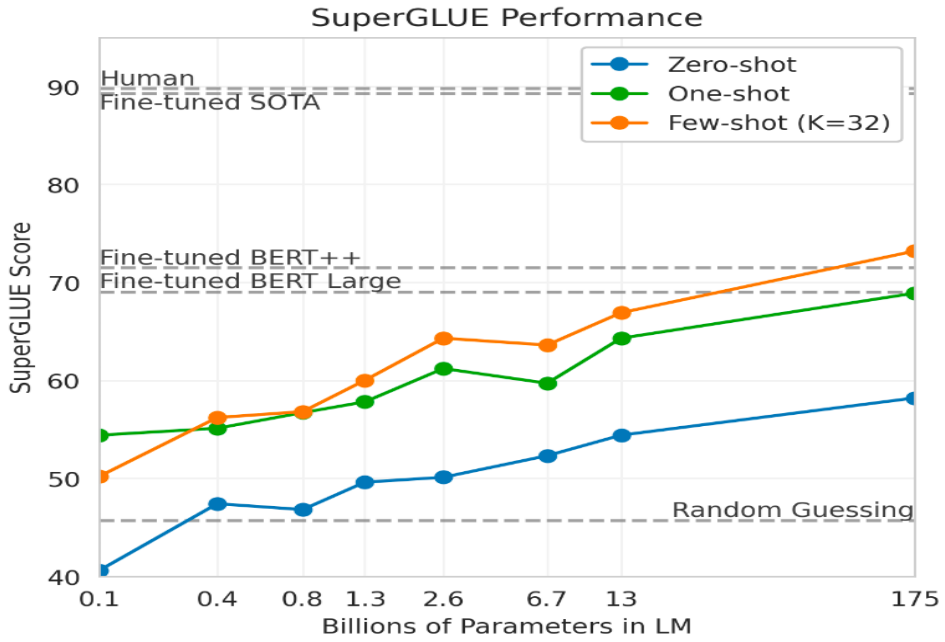
百度智能云

以OpenAI GPT-3为例

1750亿模型带来了质的效果飞跃



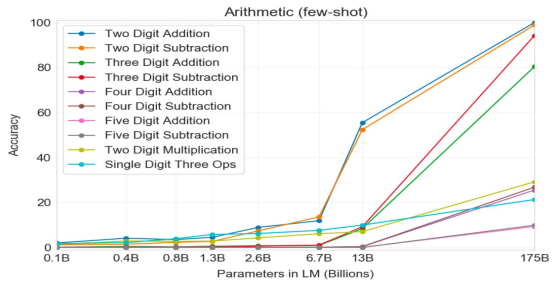
32条样本即可达到BERT的效果



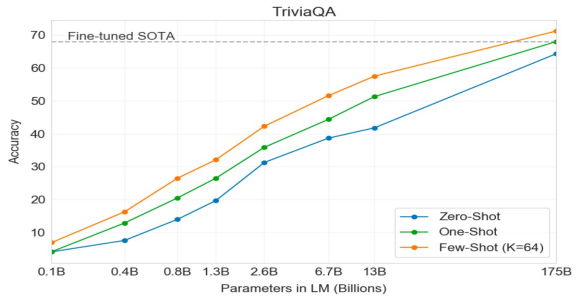
大模型带来AI通用性显著提升

百度智能云

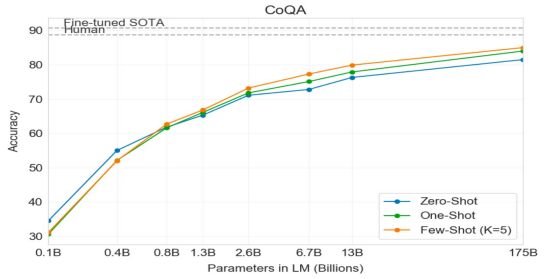
超大规模模型逐渐具备使用零样本或少量样本处理各种新任务的通用能力



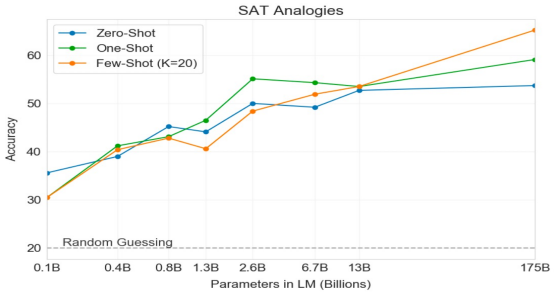
数学计算



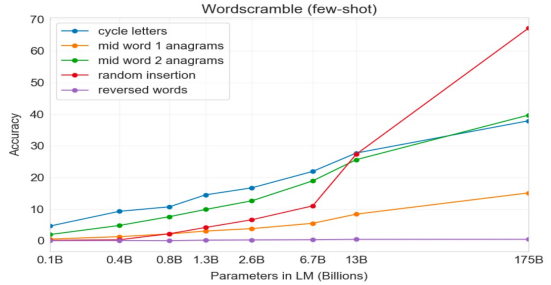
阅读理解



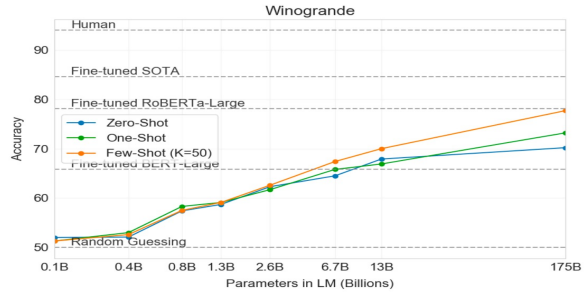
多轮问答



SAT 考试



新词构造



指代推理

注：基于GPT-3的实验分析

爆款应用拉动大模型训练需求

百度智能云

AIGC文生图

How to learn a language easily?

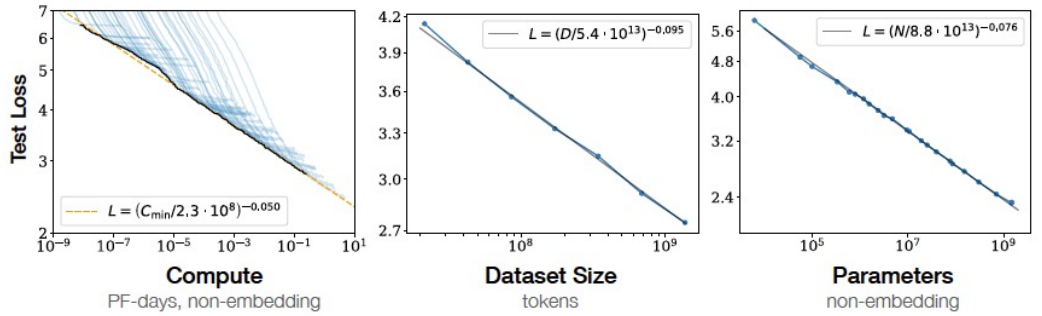
Learning a language can be a challenging but rewarding task. To make it easier, here are a few tips:

1. Start with the basics. Focus on learning the most commonly used words and phrases first. This will help you build a foundation for the rest of your language studies.
2. Practice regularly. Consistent practice is key to making progress in any skill, and language learning is no exception. Try to practice every day, even if it's just for a few minutes at a time.
3. Use real-life materials. Try to find authentic materials such as books, magazines, and websites in the language you're learning. This will help you learn how the language is used in everyday situations.
4. Talk to native speakers. If you have the opportunity to talk to native speakers, take it! This will help you improve your pronunciation and learn colloquial expressions that you might not find in a textbook.
5. Don't be afraid to make mistakes. Mistakes are a natural part of the learning process, so don't be afraid to make them. Instead, use them as an opportunity to learn and improve.

ChatGPT

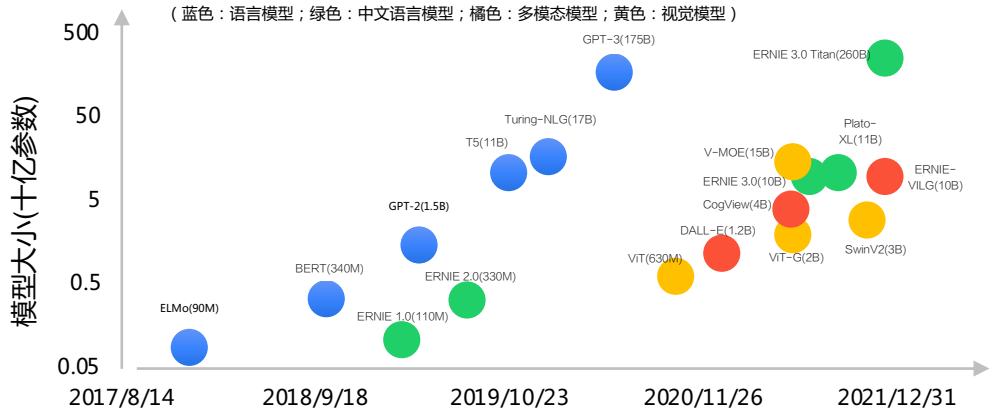
大模型训练需要足够数据与算力

百度智能云



Jared, et al. 2020.

Scaling Law三要素



模型参数演进

以GPT-3为例，1750亿参数模型、3000亿词语，计算量314ZFlops

02

超大模型训练对基础设施的需求

面向大模型的基础设施全景图

百度智能云

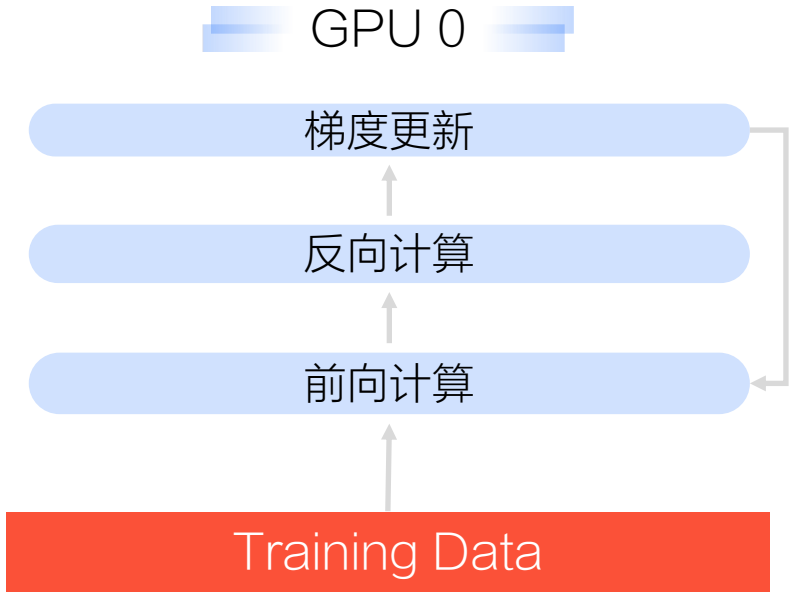
超大模型	ERNIE 3.0 Titan	VIMER-UFO 2.0	ERNIE 3.0
AI框架	Fleet + PaddlePaddle	DeepSpeed / Megatron + PyTorch	
加速库	AI算子库(cuDNN, cuBLAS、AIAK-OP)	集合通信库(NCCL, ECCL)	
资源管理层	AI平台		
	AI调度器	存储服务	
硬件资源	GPU / 昆仑	RDMA	存储节点

从框架到集群，大模型软硬结合的全栈基础设施

从AI框架入手，解决大模型的技术挑战

百度智能云

传统训练：小模型、小样本，单卡训练



大模型的变化：参数量与计算量激增

ZFLOPs计算量

GPT-3: 314 ZFLOPs

大规模参数

175B 参数

算力墙

- A100算力312TFLOPS
- 单卡需要32年
- 需要分布式加速

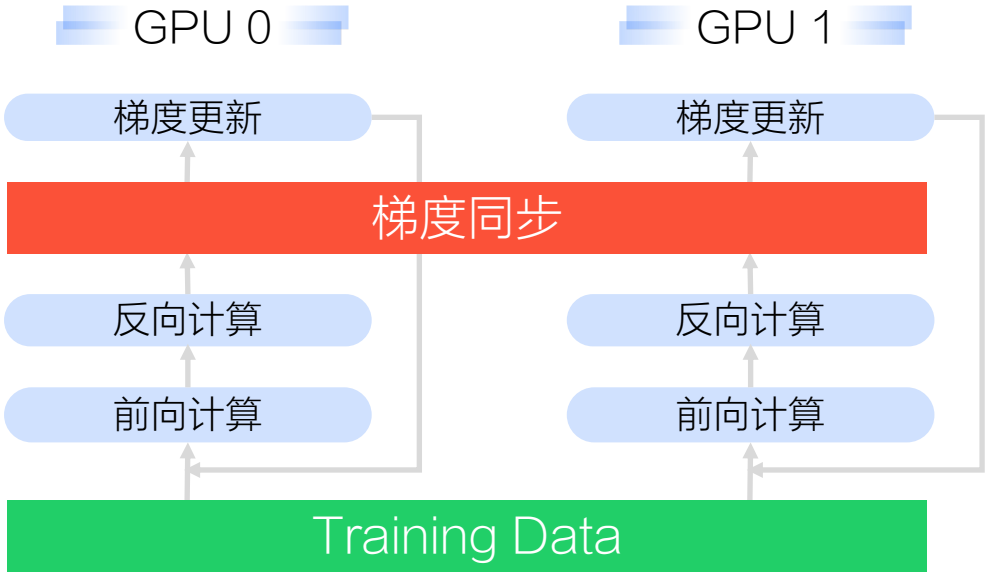
存储墙

- 千亿参数需要2TB存储
- 单卡显存80GB，放不下
- 需要更多存储空间

对模型和数据进行切分

算力墙 —— 数据并行

百度智能云



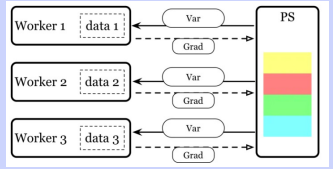
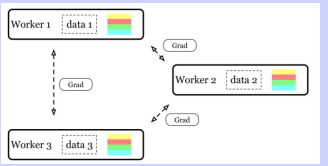
数据并行：

- 对数据集进行切分，不同卡模型相同，数据不同
- 由于数据不同，所以不同卡计算的梯度也不同
- 为了迭代中多卡参数一致，需要引入梯度同步
- 梯度同步过程即多卡把各自梯度求平均的过程

数据并行中主要研究方向就是梯度同步，常见评价指标如下：

- 加速比 = 多卡全局吞吐 / (单卡吞吐 * 卡数)
- 收敛性 = 精度收敛到一定范围的时间

常见梯度同步策略：同步更新 vs 异步更新

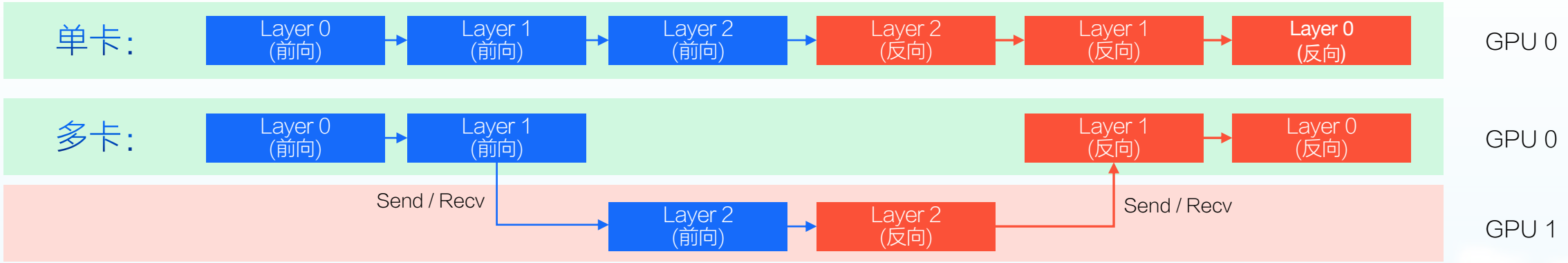
策略	异步更新	同步更新
实现	节点异步上报局部梯度，更新并获取全局梯度，不等待其他节点 	节点间阻塞等待，同步上报局部梯度，并同步更新全局梯度，常用AllReduce实现 
加速比	无阻塞，100%	结合通信重叠等优化，在高性能网络下，可以做到95%+
收敛性	存在梯度滞后、部分更新等问题	收敛较稳定

目前大模型训练主要采用同步更新策略

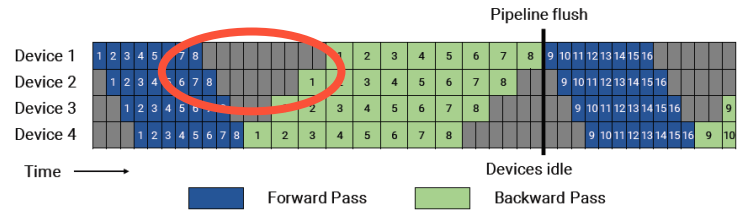
存储墙 —— 流水线并行

百度智能云

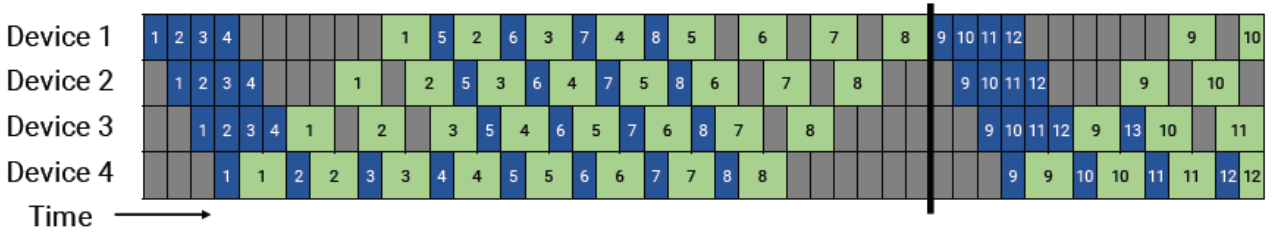
按层切分：每张卡保存部分层，通过点对点Send/Recv同步激活与梯度；将数据切分成mini-batch传入流水线



流水并行的主要问题：流水线气泡



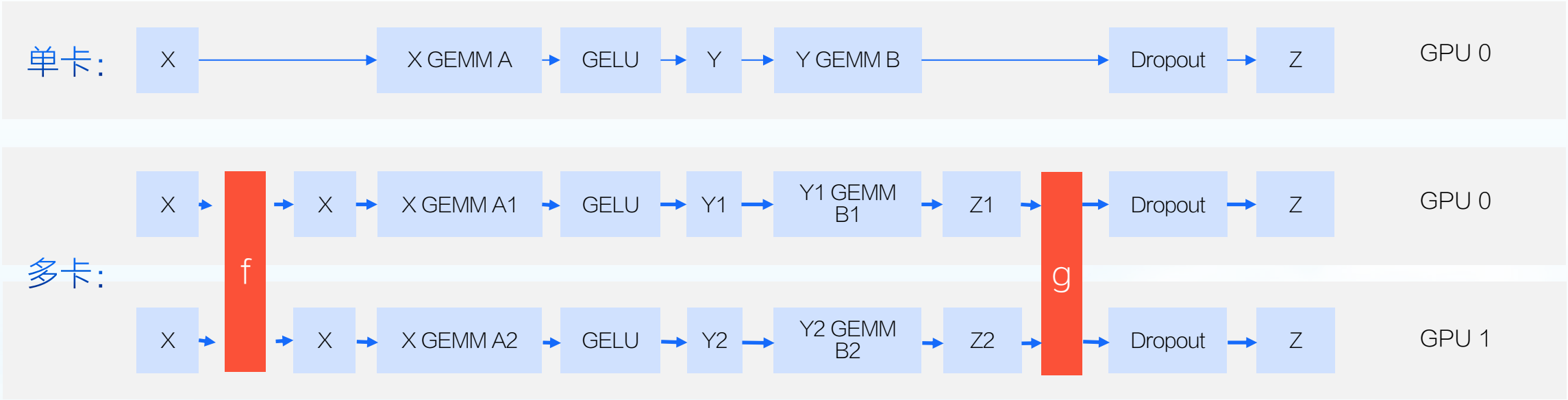
mini-batch 1反向需要等前向完成，导致Device 1空等



通过调整不同mini-batch数据执行顺序，减少气泡

存储墙 —— 张量并行

层内切分：对于单层参数仍然过大问题，可以将单层操作切分到多卡进行



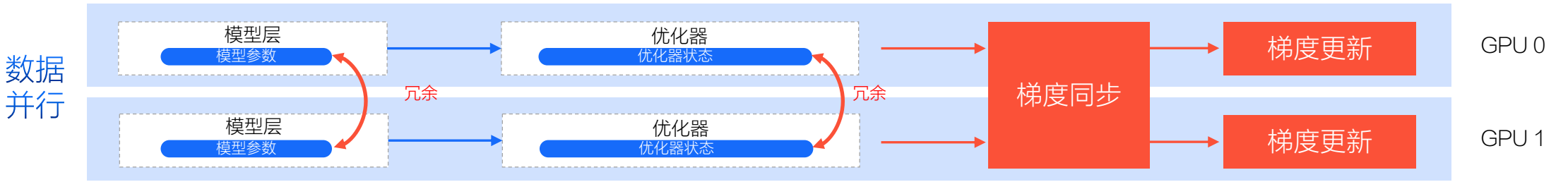
f = Identity g = AllReduce

把GEMM操作的权重切分，每张卡处理一部分矩阵乘结果，
最后通过AllReduce汇聚结果

存储墙 —— 分组参数切片

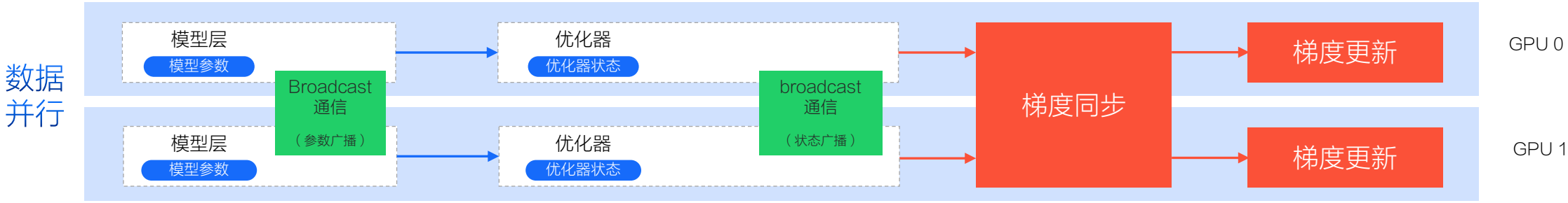
百度智能云

数据并行的显存冗余： 数据并行中的每张卡都会保存一份完整的模型参数、梯度及优化器状态



在每次梯度同步后，多卡上的模型参数、优化器状态会保持一致，造成存储上的冗余，浪费显存

分组参数切片 将参数与优化器状态在参与数据并行的卡间切分，计算时按需通信同步，时间换空间



假设有N卡参与数据并行，每卡显存可节省为 1 / N

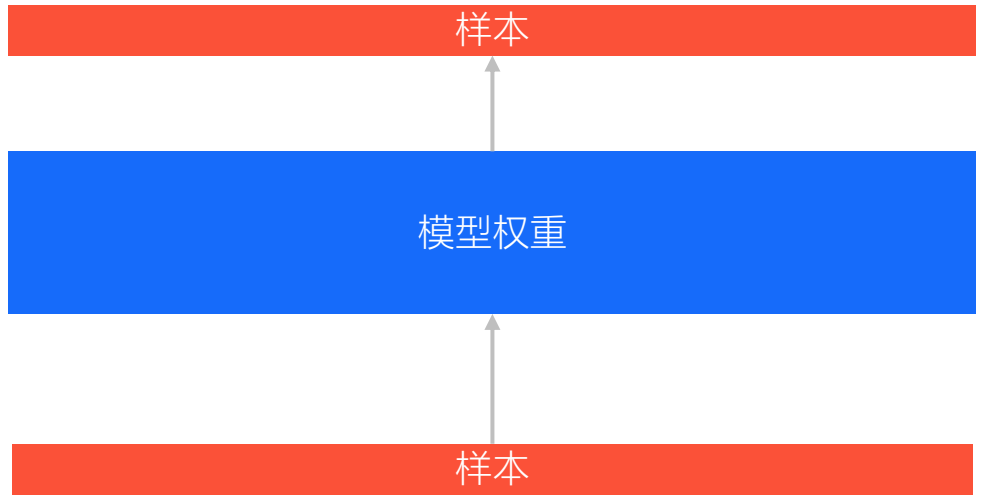
大模型加速 —— 减少计算量

百度智能云

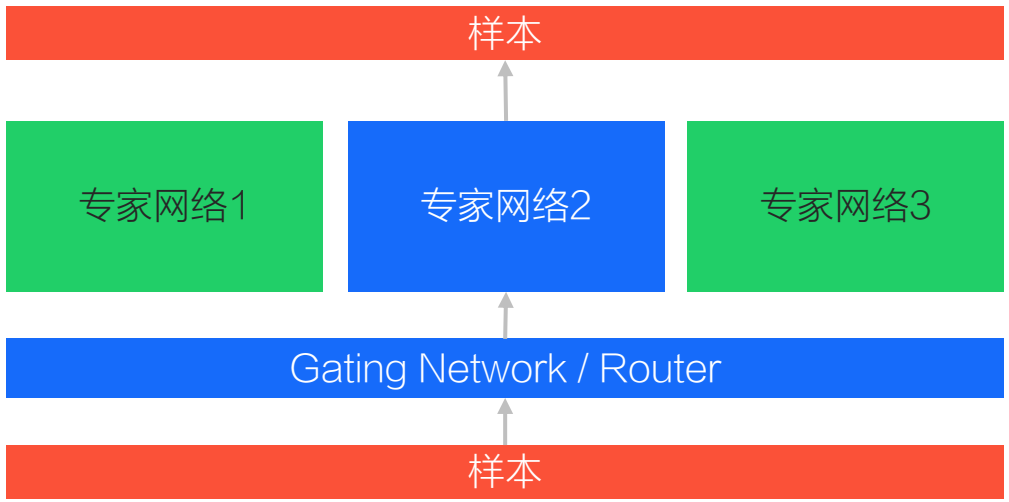
当数据量足够大时，参数越多的模型精度越好；而参数量增加造成计算量增加，需要更多资源

关键问题：如何保证参数规模的同时，减少计算量?

解决方案：条件计算，根据条件（路由）激活部分参数；将模型参数拆分成多个子网络（专家网络）



单样本激活所有参数



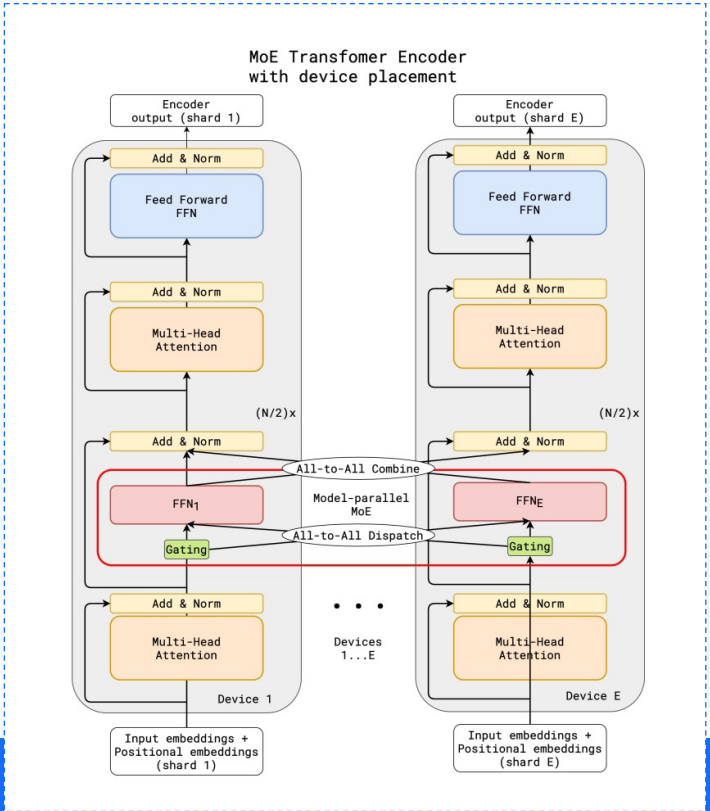
单样本按条件激活部分参数

（上图中单样本计算量减少1/3）

减少计算量 —— 混合专家模式

百度智能云

混合专家：基于条件计算范式，将模型抽象为多个专家，每卡处理不同的样本，并独立计算路由



- 每张卡处理不同的数据分片（ shard 1 .. shard E ）
- 在Gating计算过程考虑所有设备
- 每张卡的样本可能被其他所有卡计算，同时可能接受其他卡的样本
- 通过全局All2All操作将数据放置到对应的设备
- 同参数量的模型，效果不如混合并行策略的模型

（序号代表Gating后适合放置的卡；颜色表示当前所在的卡）

并行策略实战 —— 飞桨4D混合并行训练

百度智能云

大模型，堆叠 Transformer 层，天然适合切分

竖切、横切、纵向扩展

竖切
按 Transformer 层切分，称为流水线并行（PP）

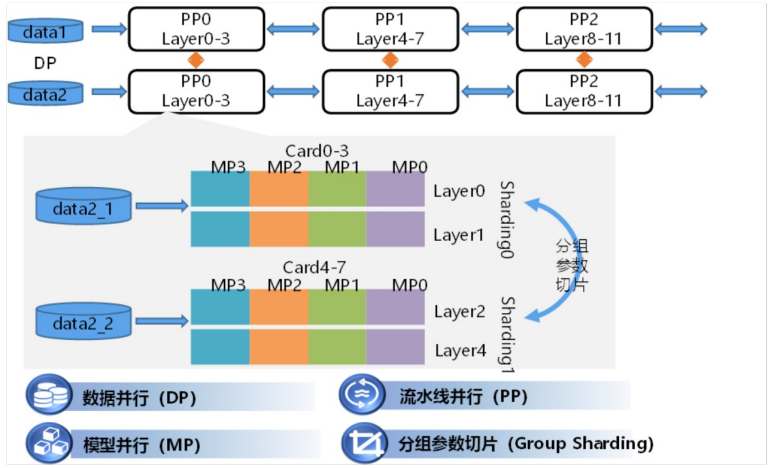
横切
Transformer层内大MatMul切分，称为模型并行（MP）

纵向扩展
将训练数据切分加速训练，称为数据并行（DP）

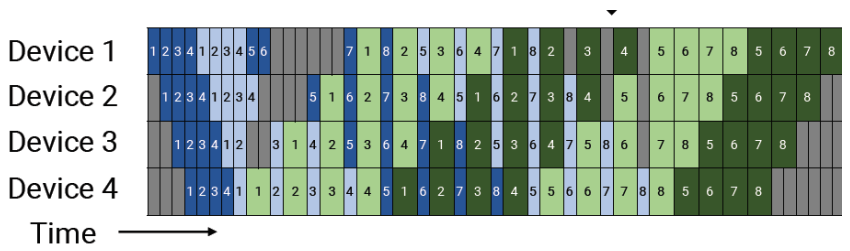
模型切分
模型参数分组，减少显存占用（Sharding）

千亿模型训练配置

策略	参与节点
张量并行	机内8卡AllReduce
Sharding	机内8卡Broadcast
流水并行	多机一组，机间同号卡P2P通信
数据并行	多组数据并行（多机一个单元）



飞桨 4D 混合并行框架



高效流水并行编排

硬件资源 —— 大模型训练对算力和通信的需求

百度智能云

切分方式	通信操作	通信量（单卡）	通信卡数	计算时间	对集群的需求
模型并行（MP）	AllReduce	百 GB/PP 数量	单机 8 卡	秒级	机内高速互联
流水并行（PP）	Send / Recv	MB 级别	多机 2 卡	秒级	P2P 低延迟
数据并行（DP）	AllReduce	GB 级别	全部卡，分多组	十秒级	高吞吐 AllReduce
参数分组（Sharding）	Broadcast /AllGather	百 GB/PP 数量	单机 8 卡	秒级	机内高速互联
专家并行（MoE）	All2All	百 GB	百卡	秒级	高集群整体网络吞吐

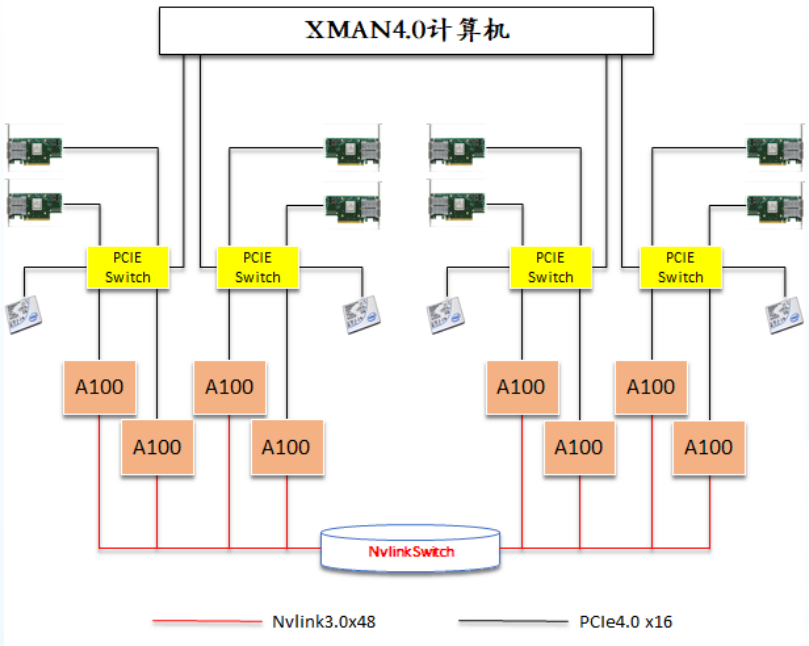
预估：1750 亿参数、3000 亿词语、1024 卡 A100，需要 34 天训练

单机硬件选型

百度智能云

- 设计目标：算力高、机内多卡通信能力强
- 设计折衷：机内拓扑设计

GPU	8x NVIDIA A100 80GB
显存	80GB * 8
NVSwitch	6
CPU	2S/4S
网卡	8个端口 200Gb/s
存储	4T * 8 NVME SSD
性能	5 petaFLOPS @ FP16
机内互联	134GB/s (AllReduce算法带宽)



集群网络设计

百度智能云

通信需求：大模型集群可达万卡级别，单作业千卡；兼顾P2P延迟和通信吞吐

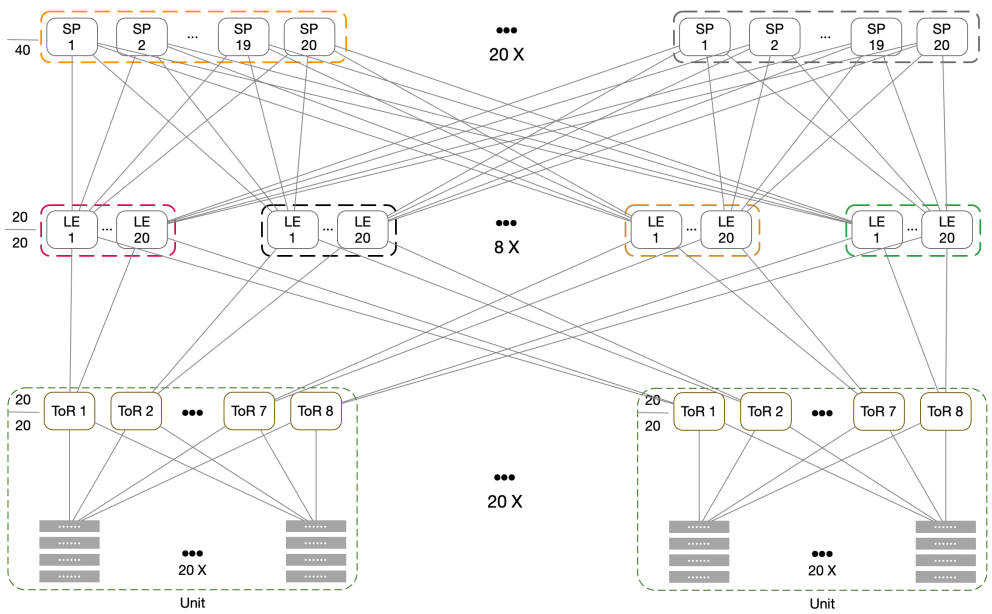
通信特点：AI训练中网络侧最多的流量是同号卡AllReduce操作

8导轨优化的三层CLOS架构

- 最大可支撑16000卡规模，目前IB盒式组网最大规模
- 结合网络流量特点，重点优化同号卡AllReduce操作
- 20台机器为一组（Unit）
- 一组机器有8台TOR组成，分别连接20台机器对应变化的GPU网卡
- 多组Unit间的同号卡通过Leaf层连接，支持最大400卡AllReduce互联
- 异号GPU网卡通过Spine层连接，使能异号卡网络通信

与Dragonfly、Torus拓扑比较的优势

- 网络带宽更充足
- 节点间跳步数更稳定



网络拓扑设计图

百度智能云

03

软硬件结合的联合优化

基于静态图的多后端加速架构

百度智能云

Graph Capture

Abstract Backend Layer

FastDeploy

Graph Optimizer

Graph Converter

TensorRT

TVM

基础算子库
(如cuDNN)

AIAK-OP
(代码自动生成)

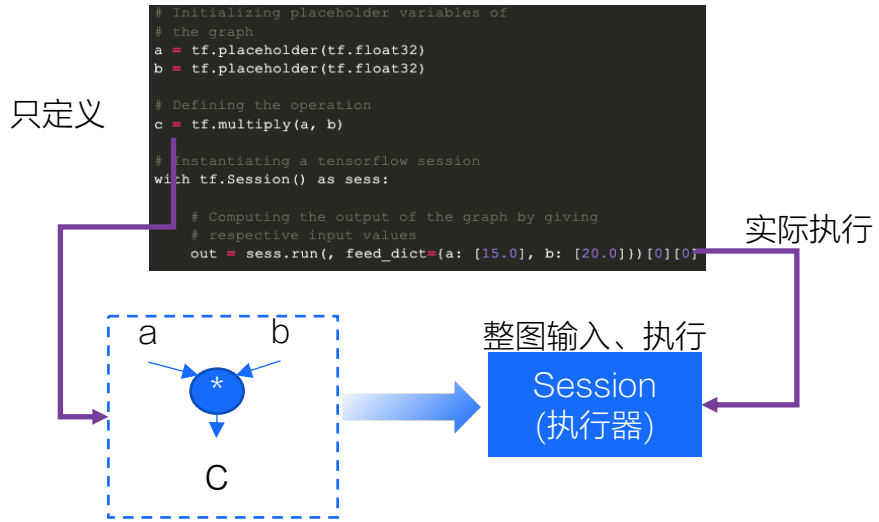
- ✓图接入：框架动态图到静态图捕获组件
- ✓后端抽象：支持多后端接入，提供基于计时的选优能力
- ✓图优化：通用图优化(DCE、CSE)、重点算子融合
- ✓图转换：将聚类后的图转换为后端表示、加速算子库使能
- ✓多后端：兼容手写+编译的多种后端，提供可扩展后端抽象
- ✓算子库：针对典型模型结构专门优化；基于Cutlass模块化算子生成；基于TVM算子自动编译优化

大模型加速 —— 图接入

百度智能云

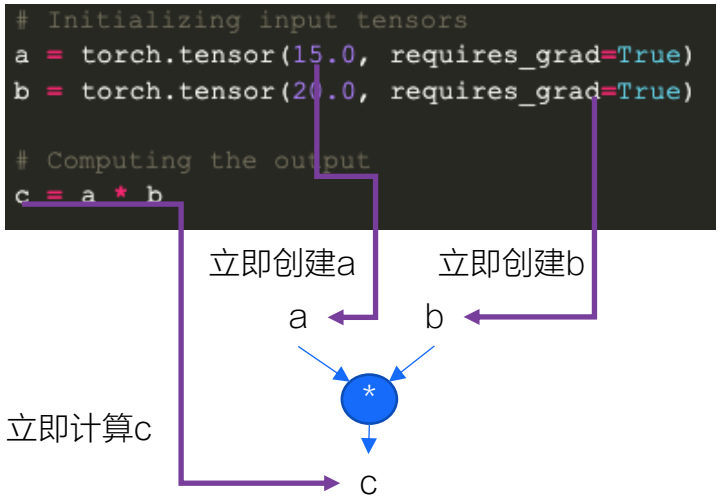
AI框架提供一系列API进行模型图描述，按图执行时机分为动态图与静态图

静态图：先定义后执行（define and run）



- 专有API构建图、异步执行
- 整图执行前无法获取值，难开发、难调试
- 执行器有整图信息，有较好的性能优化空间

动态图：图构建与计算执行同时发生（define by run）



- 每执行一条Python语句就立即求值（异步）
- 中间计算结果可随时获取，容易开发与调试
- 执行器每次只看到一个小操作，不易优化性能

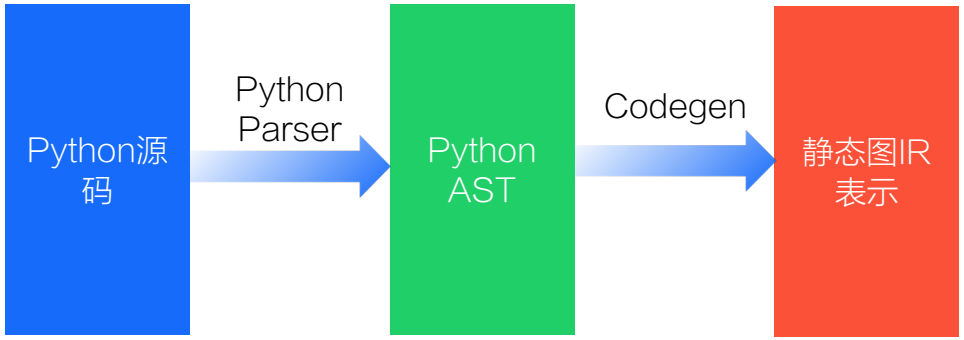
动态图易于开发调试，静态图易于优化执行，算法工程师更加喜好动态图框架（如PyTorch）

各取所长，动态图与静态图融合

百度智能云

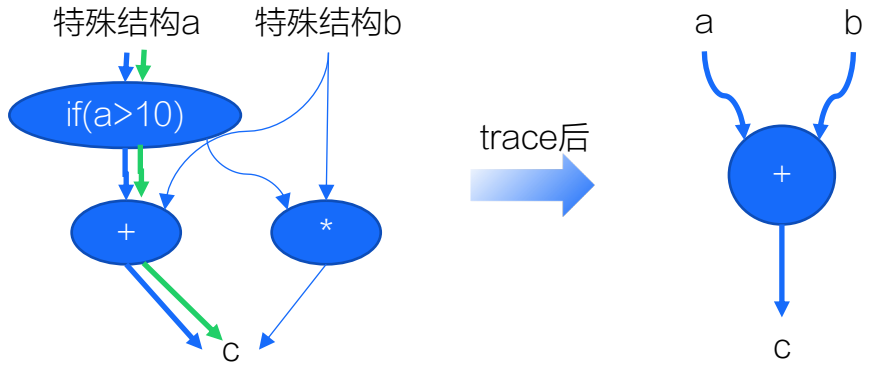
关键问题：能否使用易用的动态图开发，再通过静态图优化执行？

路线一：基于Python AST的静态转换



- 静态代码分析，从Python AST入手
- 将Python AST中的函数调用转换为静态图操作
- **Python语言灵活性导致静态分析无法理解语义**
  - 例如，静态分析无法推断动态类型
  - 又如，静态分析无法推断range范围
- 只适用于无动态类型的简单代码

路线二：Tracing & Symbolic Tracing



- 从执行过程入手，构造特殊结构，动态追踪捕获
- 特殊结构兼容Tensor接口，并能记录执行的操作
- 实际执行图，执行结束后，回放记录，形成静态图
- **对于依赖输入的分支、循环结构，存在安全性问题**
  - 例如**上图的if节点结果就只保留加法的一枝**
- 只适合于分支、循环条件不依赖输入数据的代码

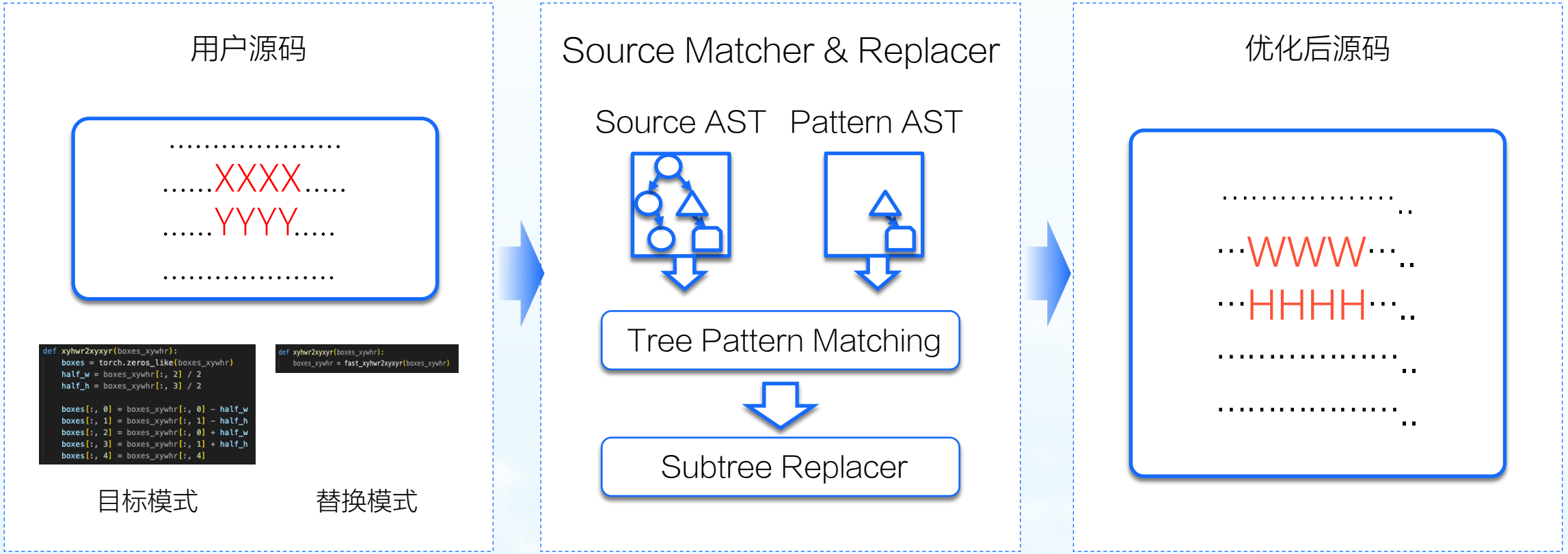
Python语言的灵活性使得动态图完整转换静态图成为（现阶段）不可能完成的任务

加速方案 —— 基于AST的代码替换

百度智能云

设计目标：用户无感的发现问题模式，替换为可trace、可编译的语义等价代码

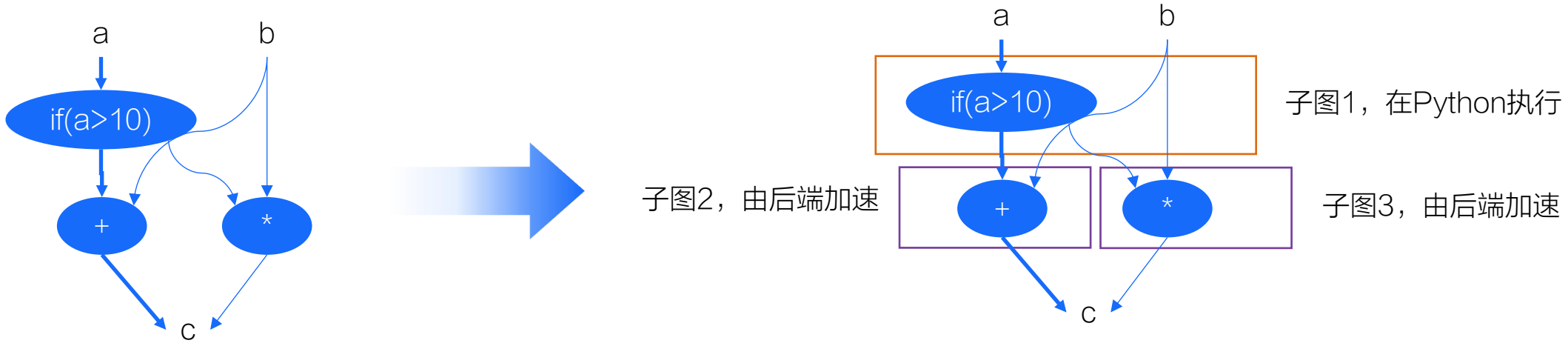
实现方案：基于Python AST的模式匹配与替换；通用方案，也可用于算子融合



社区方案 —— TorchDynamo

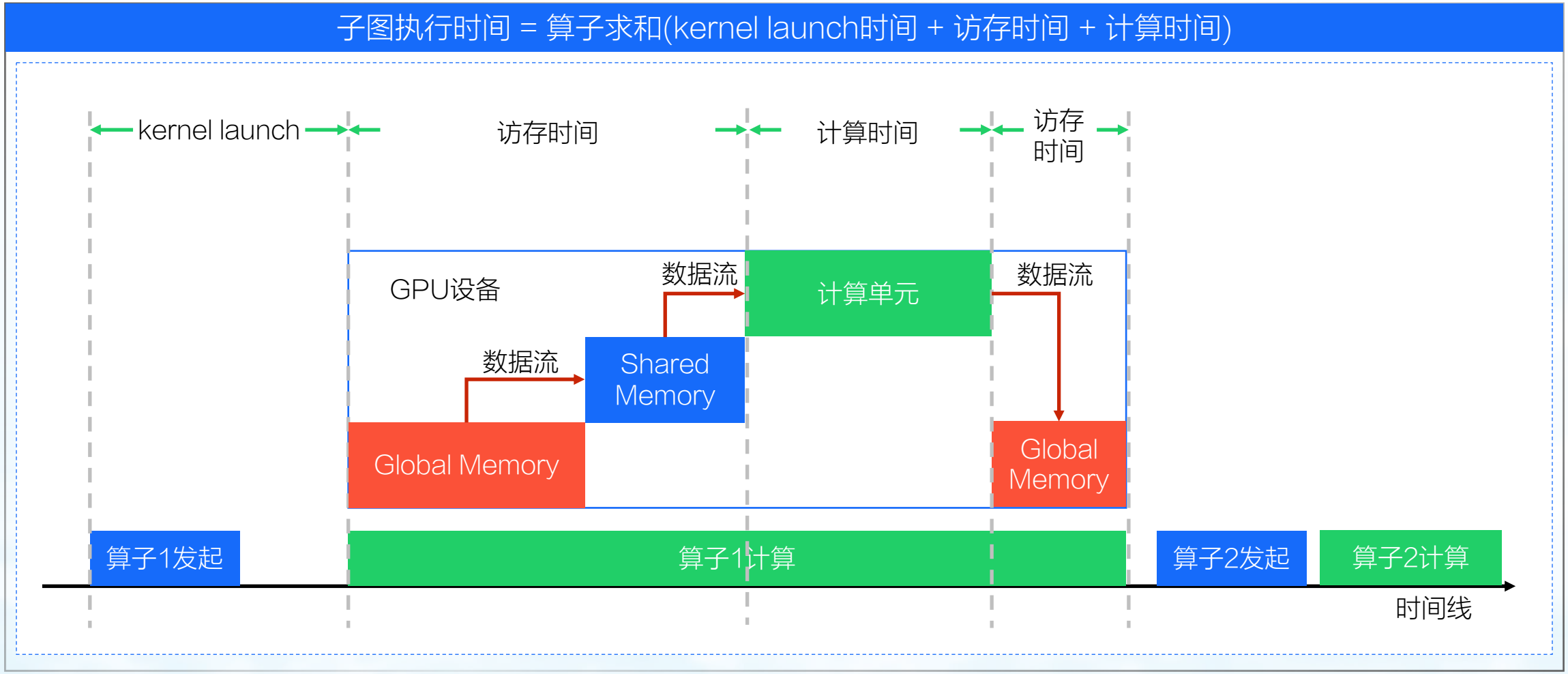
折衷方案：拥抱Python，部分捕获，不支持的结构fallback回Python语法

具体实现：基于Python Frame Evalution API，在Python Bytecode层面做劫持



社区在7500+模型上验证捕获可行性与安全性，随PyTorch 2.0发布

后端加速 —— 计算执行时间分析

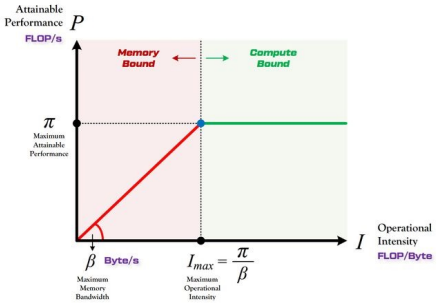


计算加速 —— 算子融合

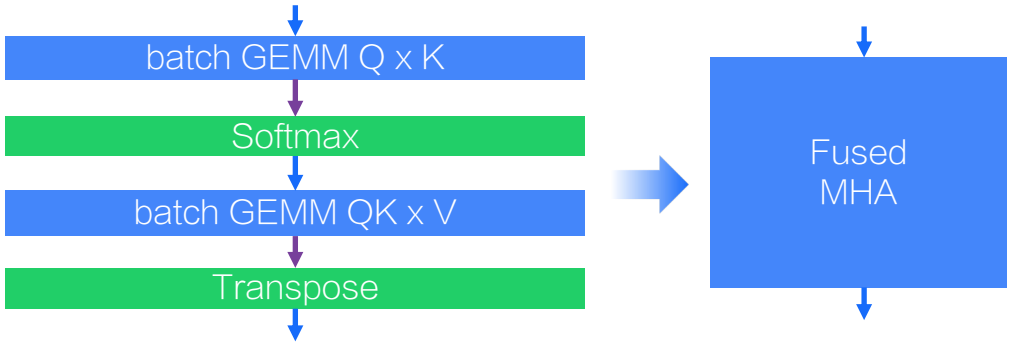
百度智能云

融合收益来源：去掉kernel launch时间，提升计算密度，减少额外访存

- 算子对单位数据上进行的计算次数，定义为计算密度
- 按计算密度根据roofline模型可以分为计算密集型和访存密集型
- GEMM通常为计算密集型算子，Elementwise通常是访存密集型算子
- 期望的目标是所有算子都是计算密集型，可以充分利用算力
- 计算密集型+访存密集型算子，访存密集型算子之间可以进行融合



融合算子举例：Multihead Attention (MHA)



计算密集与访存密集交错，适合融合

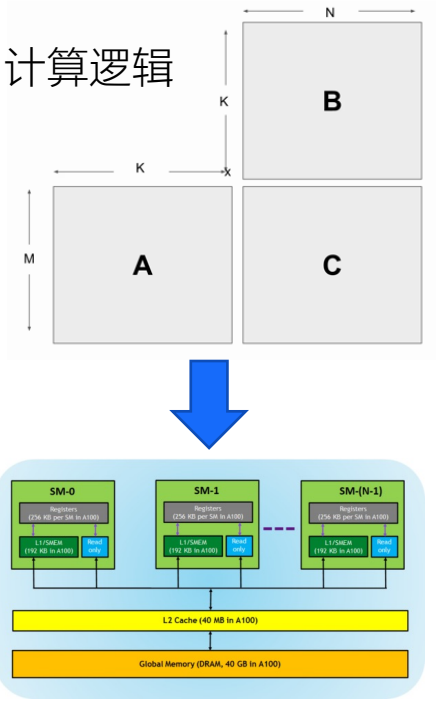
其他重要融合模式：

模型方向	算子模式
NLP	Fused MHA, SkipLayerNorn…
VIS	AdaptiveLin, YoloBox, …
通用	Conv+BN+(Mish / Gelu), GEMM + GEMM, GEMM + BN, Conv3x3 + Conv1x1, Conv1x1 + BN + HardSiLU, Conv + Reshape + Transpose + Softmax,…

计算加速 —— 算子实现优化

百度智能云

核心问题：是如何将计算逻辑与芯片架构匹配，最大化计算单元使用，降低访存损耗



芯片架构

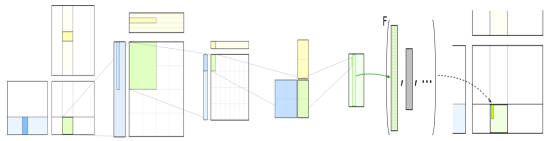
cuBLAS cuDNN

方案一：手写算子

- 如cuBLAS、cuDNN
- 芯片厂商闭源实现
- 指令级极致优化
- 主要限制：支持的操作有限且不支持二次开发

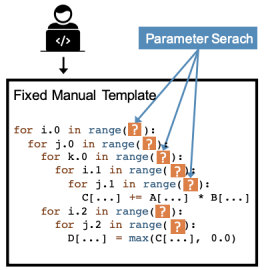
CUTLASS

CUDA Templates for Linear Algebra Subroutines and Solvers



方案二：半自动化模板

- 如CUTLASS
- 按芯片架构抽象切分循环模式
- 开源实现，可扩展
- 可参数化调优
- 主要限制：适用于解决计算密集型算子



方案三：基于搜索的优化

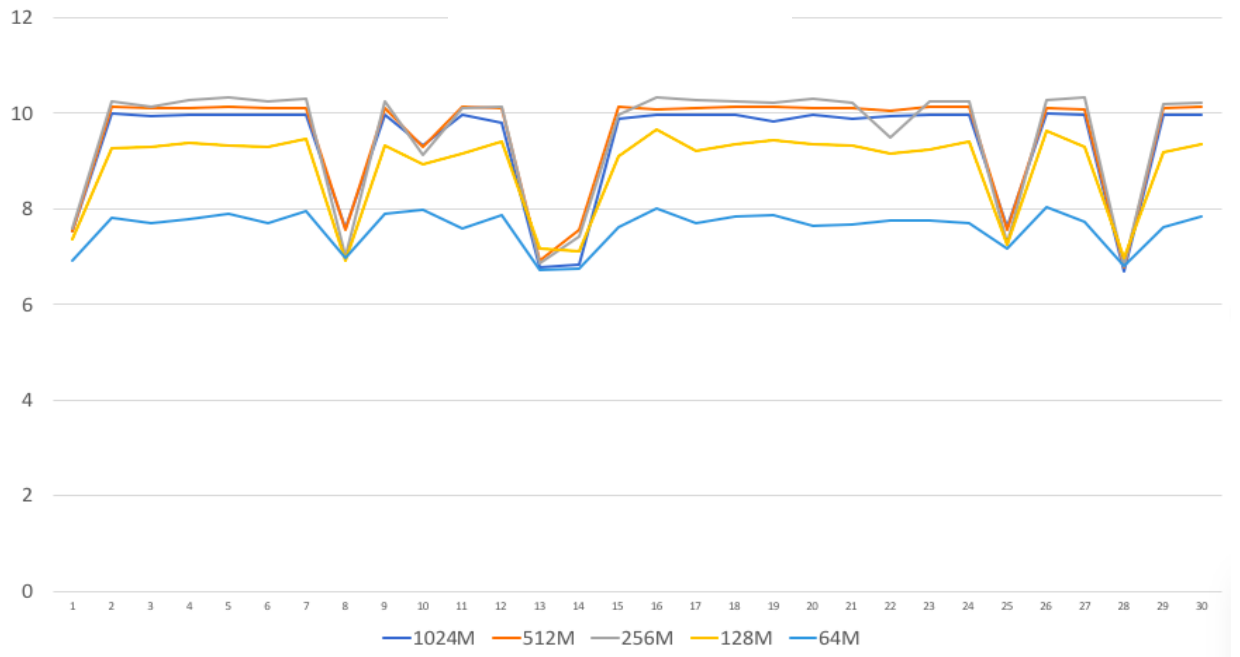
- 如Halide、TVM
- 计算与调度分离
- 计算描述操作、调度进行实现优化
- 易于生成高性能长尾算子
- 主要问题：搜索空间大，搜索不一定找到最优解

三种方案各有优劣，实践中通常三者并存，按计时选择最佳性能实现

通信优化 — 交换机哈希冲突

百度智能云

RoCE网络下交换机无收敛比，仍然可能发生网络侧流量冲突



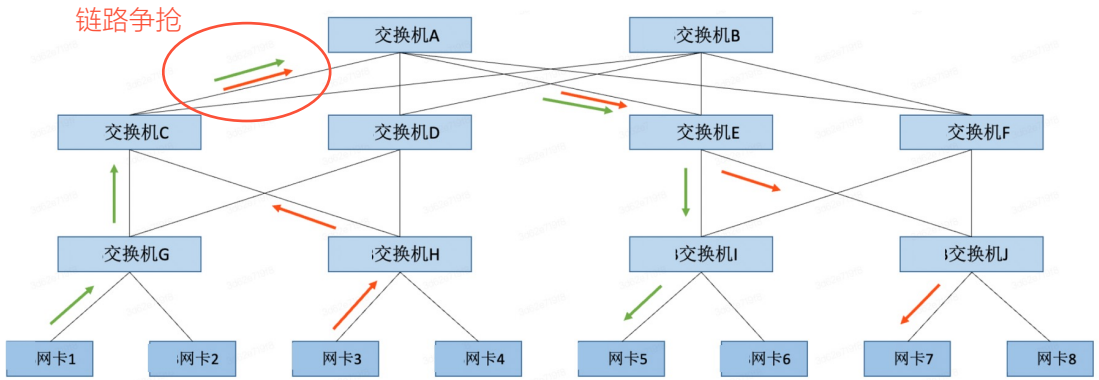
32节点30次AllReduce测试总线带宽

通信优化 — 交换机哈希冲突

百度智能云

根本原因：基于四元组的选路方式

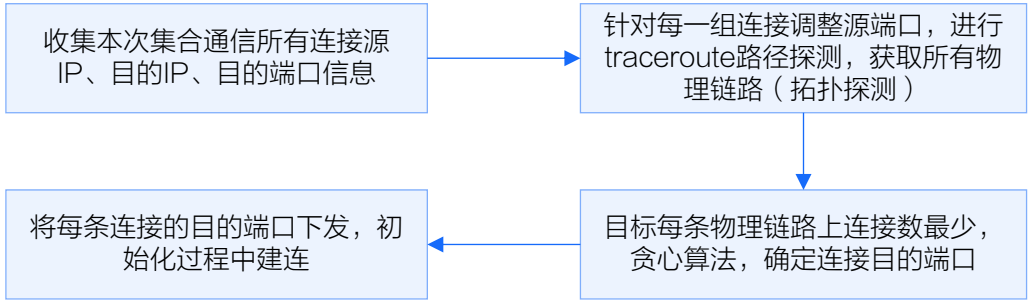
- RoCE基于以太网四元组哈希选择路径
- 当点对点有多路径可达时，可能出现多连接抢占同一链路



我们的解决方案：

- 哈希选路四元组中源端口可调，利用这一特性在建连前静态分配物理链路

（源IP，源端口，目的IP，目的端口）



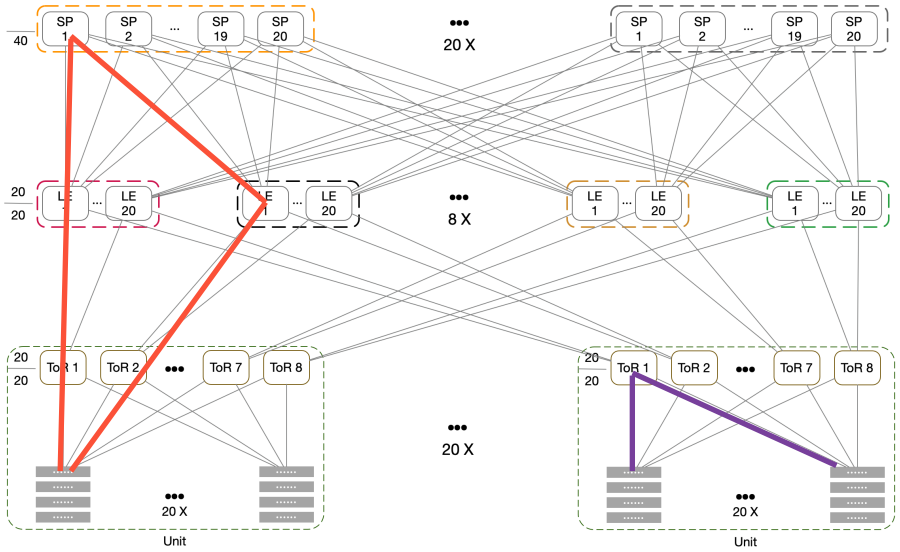
在IB协议中，也可以使用Adaptive Routing技术解决选路冲突问题

通信优化 — All2All加速

百度智能云

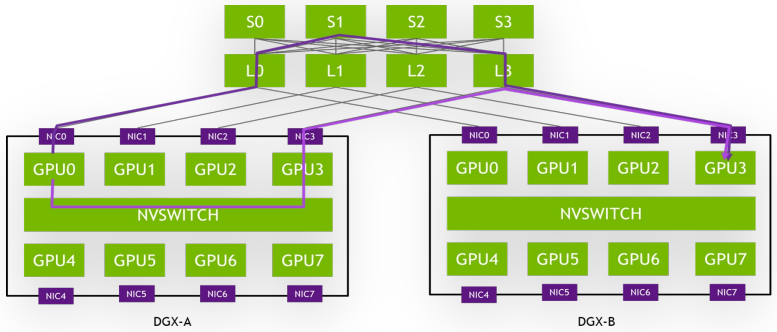
All2All加速：通过机内NVLink减轻对网络的压力

- 在8导轨优化的网络架构下，同号卡最多3条，但所有异号卡通信需要经过Spine层
- 同号卡AllReduce操作性能好，但All2All操作对网络压力较大
- 优化思路是通过机内高性能NVLink中转网络请求，从而将异号卡通信转换为同号卡通信，充分利用8导轨优势
- 如右图所示，A节点的GPU 0先经过NVLink到GPU 3，再发往B节点GPU 3



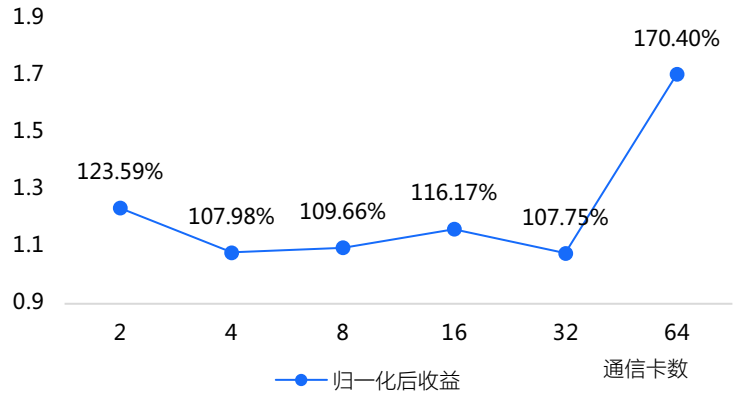
8导轨优化下，同号卡最多经过Leaf层，异号卡通信需要经过Spine层

Rail-local All2All



(NCCL 2.12+)

All2All性能提升（%）

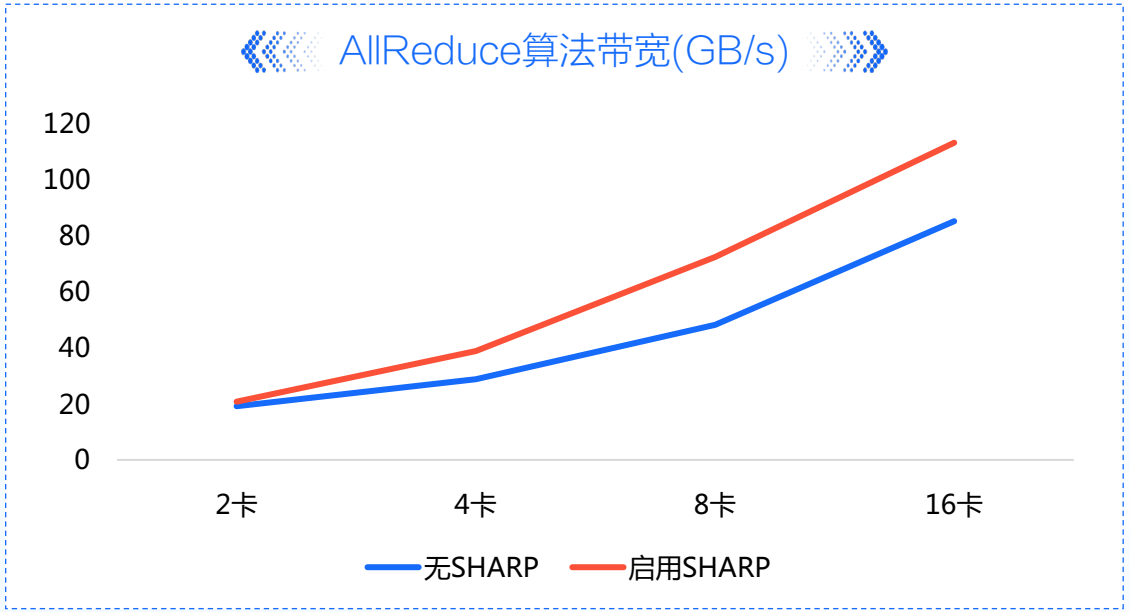


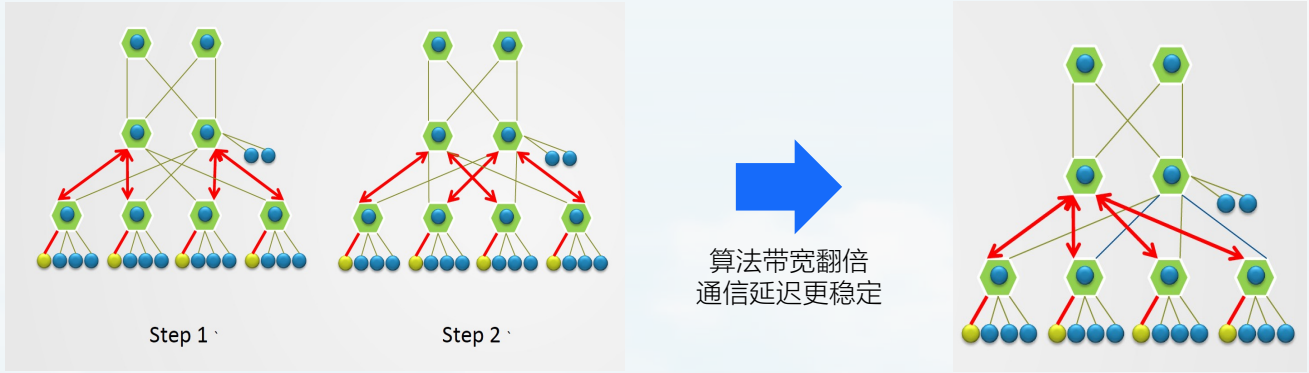
通信优化 —— 使能Infiniband

百度智能云

SHARP：网络中的计算，提升AllReduce性能

- 现有AllReduce操作通常使用GPU卡通过通信算法实现
- 常见的算法有Ring和Tree
- 无论哪种实现算法，都需要多卡间多次数据传输
- SHARP将计算能力卸载到交换机上，在数据传输中完成计算
- 核心优势：
  - ✓ 通信次数从O(logn)降低到O(1)
  - ✓ 单次数据流即可完成，算法带宽翻倍
  - ✓ GPU计算单元释放，提升计算、通信并行度



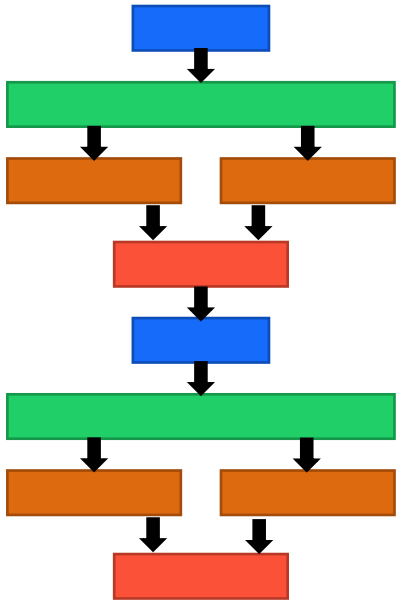


端到端自动化任务切分与放置

百度智能云

关键问题 通信策略、异构硬件共同导致手动任务切分很难找到最优解

模型代码

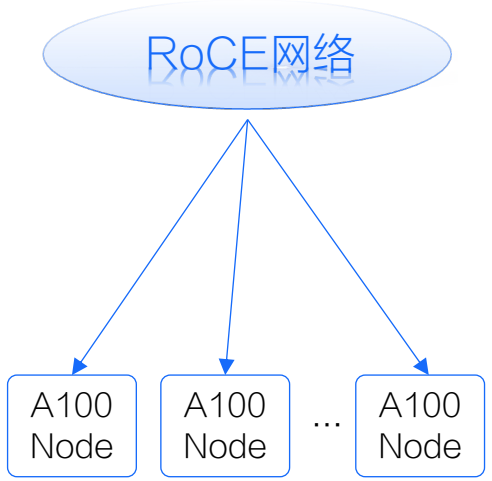


多种模型切分策略

如何放置?

物理集群

网络层级、能力描述



算力、显存、通信描述

目前基于专家经验进行切分

端到端自动化任务切分与放置

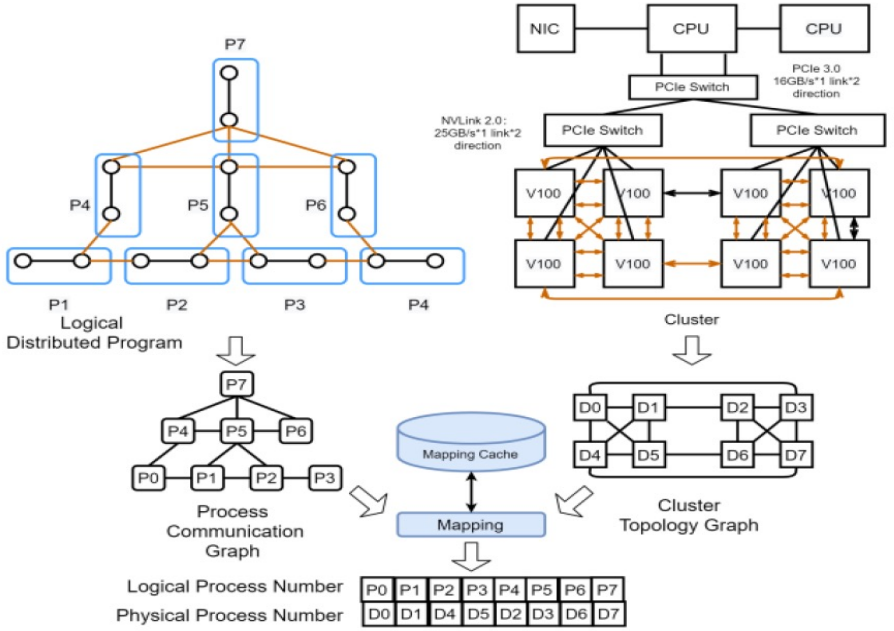
百度智能云

核心解法：

构建计算、通信的cost model，基于cost model搜索优化

- 框架侧将模型网络进行切分，并结合并行策略产出通信需求（PCG）
- 通信侧进行网络拓扑探测，结合带宽、链路构建通信cost model（CTG）
- 基于通信cost model使能框架搜索不同的切分策略，找到最优放置方案
- 找到放置策略后，再将模型图实际匹配到硬件单元上

基于cost model的自动并行方案



性能提升2.1倍

百度智能云

04

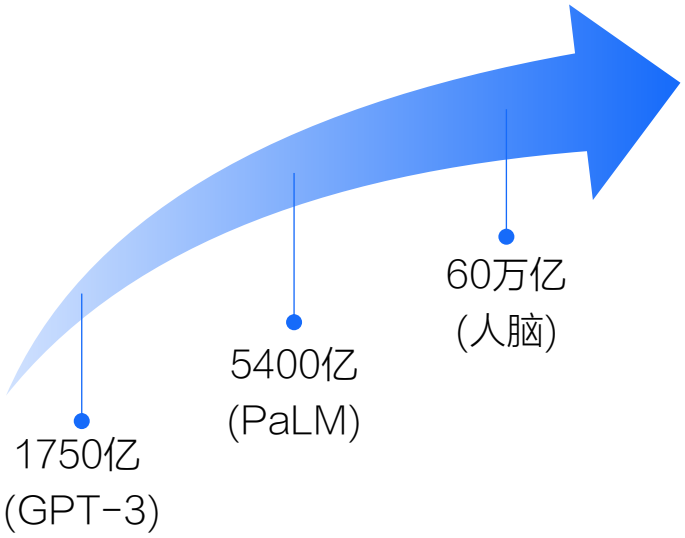
大模型发展推动基础设施演进

大模型演进的趋势

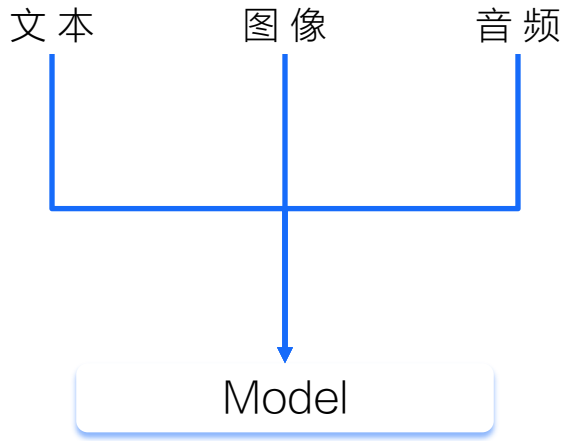
百度智能云

参数规模持续增加

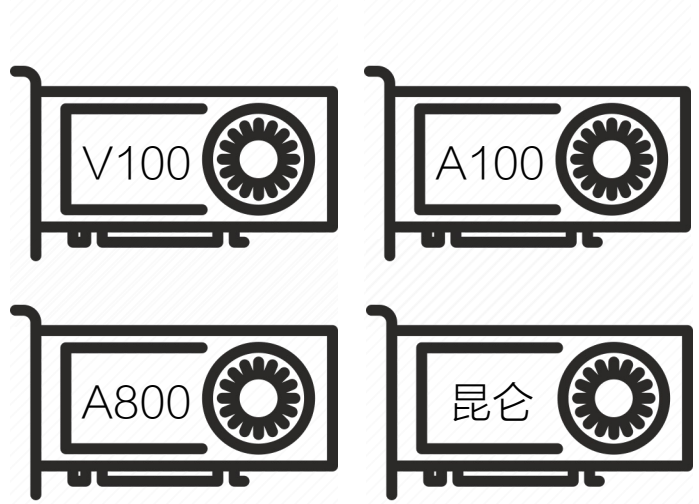
算力需求增长10000倍



多模态训练



异构资源



Scale of the human brain

集群与业务的演进

百度智能云

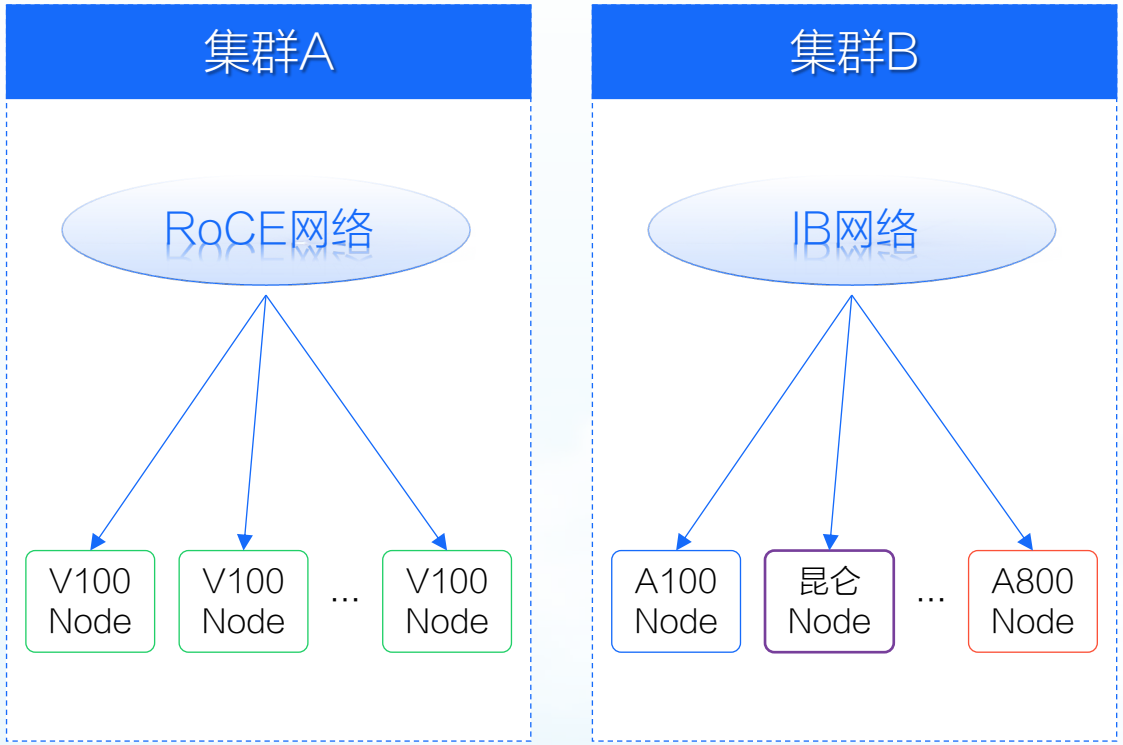
用户任务

作业1
- 100亿参数
- 1TB文本数据
- 需要1个月训练完成

作业2
- 1000亿参数
- 10TB图文数据
- 需要1个月训练完成

多类用户作业

如何放置？

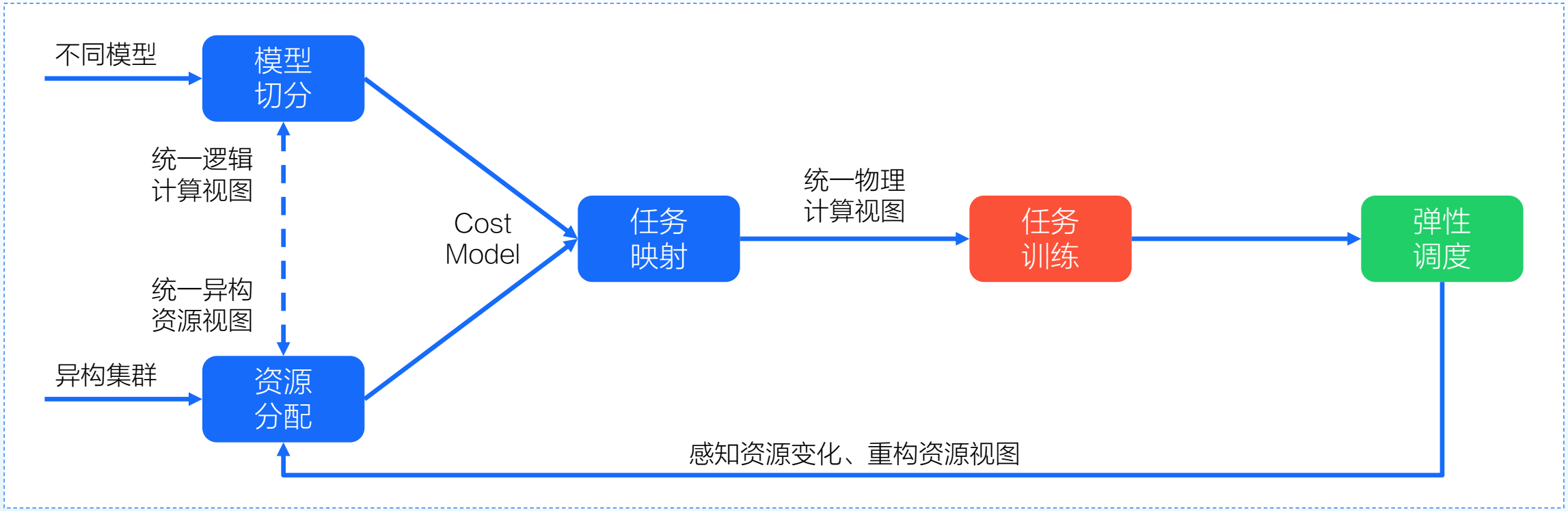


多组异构集群

基于统一视图的端到端优化

百度智能云

基于两个统一表示，来支持任意模型、资源输入，智能化自动选择最优并行策略
根据输入的动态变化，出发各个模块端到端自动的发生变化



百度百舸 · AI异构计算平台2.0

百度智能云

业务场景	城市大脑	工业互联网	产业金融	智算中心	生命科学	自动驾驶
AI容器	GPU调度	AI作业调度	弹性训练	可观测性		
AI加速	数据湖存储加速 RapidFS	分布式训练加速 AIAK-Training	推理加速 AIAK-Inference			
AI存储	海量数据湖存储 对象存储BOS	高性能存储 并行文件存储PFS				
AI计算	异构芯片 昆仑芯 / GPU	高速互联 RDMA / InfiniBand	AI服务器 X-MAN			

百度智能云

THANKS