Colossal-AI

大模型训练和部署的关键技术

尤洋

潞晨科技 (HPC-AI Tech) 创始人
新加坡国立大学 校长青年教授
加州大学 (伯克利) 计算机博士

Colossal-AI

目录
directory

- 大模型时代的机遇与挑战
- 大规模分布式训练主要技术路线
- 企业级大模型解决方案

大规模预训练模型(NLP)

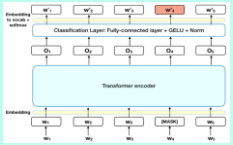
Colossal-AI

海量文本数据 → 超大规模算力 → 超大规模预训练模型 → 针对不同任务进行微调 (BERT/GPT) → 完成任务

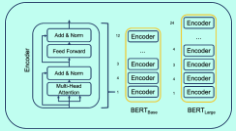
超大规模预训练模型 → 无需微调 (GPT-3) → 完成任务

主要模型

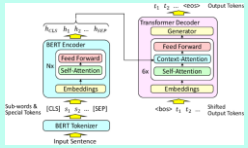
Encoder (BERT-Style) | Decoder (GPT-Style) | Encoder-Decoder (T5-Style)



BERT Architecture (Horeiv, 2018)



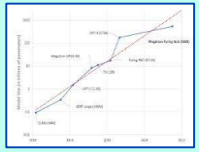
GPT Architecture (Heidloff, 2023)



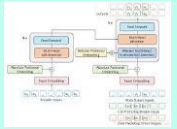
Transformer Architecture (Raffel, 2020)

研究重点

更大的模型和更多的数据 | 更高效的预训练方法 | 知识增强的预训练模型 | 小样本学习



Model Parameter (Statistia, 2022)



Model Architecture (Han, 2021)

Taxonomy of KEPTMs (Yang, 2021)

Data Augmentation (Wang, 2019)

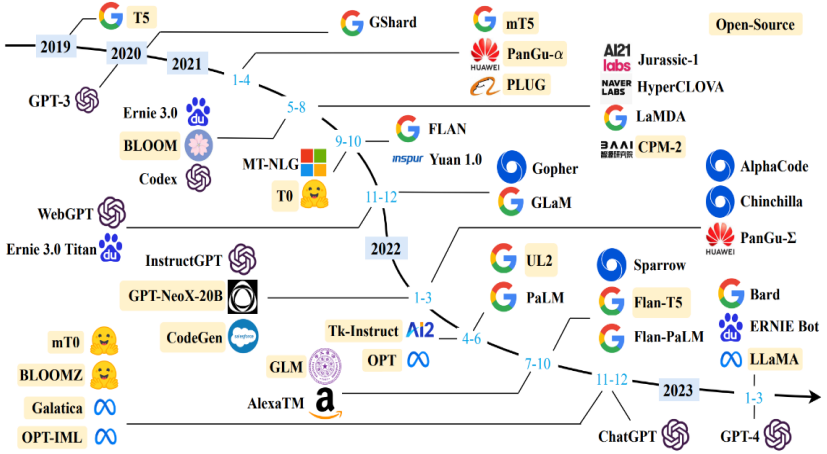


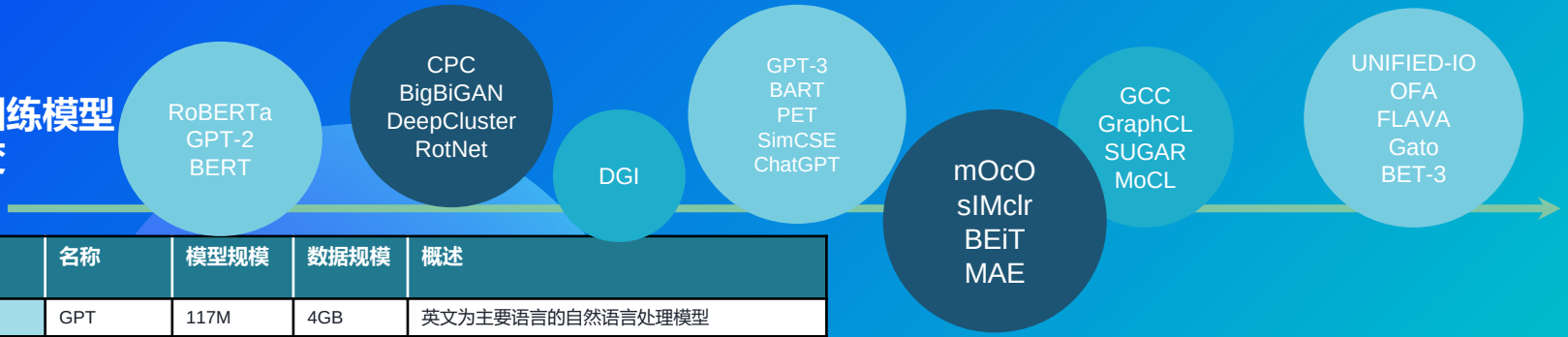
Fig. 1. A timeline of existing large language models (having a size larger than 10B) in recent years. We mark the open-source LLMs in yellow color.

大模型技术演进路线 (Reddit, 2023)

超大规模预训练模型的发展

Colossal-AI

国内外预训练模型历史和演变



时间	机构	名称	模型规模	数据规模	概述
2018.6	OpenAI	GPT	117M	4GB	英文为主要语言的自然语言处理模型
2018.10	Google	BERT	330M	16GB	英文为主要语言的语言理解模型
2019.2	OpenAI	GPT-2	1.5B	40GB	GPT迭代版本
2019.7	Facebook	RoBerta	330M	160GB	已英文为主要语言的语言理解模型
2020.6	OpenAI	GPT-3	175B	2TB	GPT迭代版本
2020.11	智源研究院	悟道 文源	2.6B	100GB	以中文为主要语言的大规模预训练语言模型
2021.1	智源研究院	悟道 文汇	11.3B	303GB	面向认知的大规模新型与训练语言模型
2021.4	阿里达摩院	PLUG	27B	1TB+	以中文为主要语言的插槽大规模语言模型
2021.4	鹏城+华为	盘古a	200B	1TB+	以中文为主要语言的全开源超大规模语言模型
2021.5	阿里达摩院	M6	1010B	1.9TB图+292G文	中文图文你多模态训练模型
2021.6	智源研究院	Wudao 2.0	1.75T	4.9TB图+文	图文多模态稀疏与训练模型
2022/4	Google	PaLM	540B	780B tokens	以英文住主要语言是的语言模型

大模型特点

➢ **参数规模大**

预训练模型突破到了亿这个级别，在此后的模型参数的规模增长呈现了一种指数级的跨越式增长，并且能够实现效果的持续提升。

➢ **数据规模大**

2018年的 BERT模型，使用了BooksCorpus (单词量 800M)，English Wikipedia (单词量 2,500M)进行训练，总体数据量在GB级别，然而到了中文领域，数据量直接飙升到了TB 起步，其中悟道2.0用了3TB 数据，ERNIE3.0用了4TB 数据。

模型发展现状分析

Colossal-AI

https://www.youtube.com/watch?v=tgB671SFS4w

训练一个LLaMA类模型所需的数据量

Colossal-AI

从零开始预训练大模型的起点

这些数据集都是公开的

数据量非常大

https://github.com/togethercomputer/RedPajama-Data/tree/main/data_prep

Dataset	Token Count
Commoncrawl	878 Billion
C4	175 Billion
GitHub	59 Billion
Books	26 Billion
ArXiv	28 Billion
Wikipedia	24 Billion
StackExchange	20 Billion
Total	1.2 Trillion

国内大模型汇总盘点

Colossal-AI

类别	厂商	大模型名称	数据规模	模型简介
互联网巨头	阿里云	通义千问	10万亿	阿里云开发的聊天机器人，能够与人互动、回答问题及协作创作。
	腾讯	混元	万亿	「混元」系列AI 大模型覆盖了NLP、CV、多模态等基础大模型以及众多行业/ 领域大模型。
	华为	盘古	1000亿	华为即将上线的“盘古系列AI大模型”分别为NLP（自然语言处理）大模型、CV（机器视觉）大模型、科学计算大模型，都已经被标注为即将上线状态。
	京东	言犀	千亿	言犀是融合京东自身十年客户服务与营销的最佳实践以及自研全链路AI能力的服务数智化平台级产品
	网易	伏羲	110亿	网易伏羲是网易旗下专业从事游戏与AI研究和应用的顶尖机构。专注数字孪生、强化学习、用户画像、NLP、分布式引擎等多领域AI技术创新，提供瑶台沉浸式虚拟活动平台
	百度	文心一言	100亿	百度全新一代知识增强大语言模型，能够与人对话互动，回答问题，协助创作，高效便捷地帮助人们获取信息
服务器龙头	浪潮信息	源1.0	2457亿	单体模型参数量超越美国OpenAI组织研发的GPT-3模型，成为全球最大规模的中文语料AI巨量模型。
AI公司	达摩院	八卦炉	174万亿	自研大模型从芯片到软件全覆盖聆心智能超拟人大模型
	达观数据	曹植	500亿	基于长期的NLP实践和海量数据积累推出国产版ChatGPT模型“曹植”系统
	云从科技	行业精灵	百亿	旨在提升公司在人机协同操作系统认知层面的能力，通过行业专家知识与大量多维度的数据训练
	商汤科技	书生2.5	30亿	图文跨模态开放任务处理能力可为自动驾驶、机器人等通用场景任务提供高效精准的感知和理解能力支持。
	毫末智行	DriveGPT	7。74亿	是第一款智能驾驶大模型，可对标 GPT-2。
	科大讯飞	1+N认知智能	/	“1”是通用认知智能大模型算法研发及高效训练底座平台，“N”是应用于教育、医疗、人机交互、办公、翻译、工业等多个行业领域的专用大模型版本。同时，“N”个场景的示范性应用产品也将随之呈现。
	IDEA研究院	二郎神	/	中文预训练语言模型“二郎神”在中文语言理解权威评测基准FewCLUE 榜单上登顶。
	聆心智能	超拟人	/	具有可控、可配置、拟人特点的大模型，通过简单设置即可构造一个有知识、有个性、有风格的类人智能体。
	竹间智能	魔力写作	/	竹间智能通过Emoti Salesmate为一家头部车企部署智能销售助手，通过对销售人员的语音会话进行解析，生成智能洞察。
科研院所	智源研究院	悟道2.0	1.75万亿	全球最大的超大规模智能模型“悟道2.0”
	中科院自动化研究所	紫东太初	千亿	“紫东太初” 跨模态通用人工智能平台以多模态大模型为核心，基于全栈国产化基础软硬件平台，可支撑全场景AI应用。
	复旦大学	MOSS	175亿	由复旦大学自然语言处理实验室邱锡鹏教授团队开发。
	清华大学	ChatGLM	62亿	基于GLM的实现方案，其6B模型已公布权重。
	上海人工智能实验室	风乌	/	“风乌”提供了一个强大有效的全球中期天气预报的AI框架，其领先性体现在预报精度、预报时效和资源效率三方面。

为什么需要1000个GPU?

Colossal-AI

内存开销大 -> 系统崩溃

GPT-3.5大约4000亿参数：假定用单精度(fp32)，至少需要8000 GB内存

- 参数占内存 1600 GB
- 梯度占内存 1600 GB
- Adam优化器一阶矩占内存 1600 GB
- Adam优化器二阶矩占内存 1600 GB
- Activation等中间结果~1600*批量 GB
- 输入数据占内存由Sequence长度和批量决定

单个A100 GPU内存：80 GB

单个DGX station GPU内存：640 GB

计算量大

GPT-3.5大约有4000亿参数
ResNet-50大约2000万参数 (很小的模型)，用ImageNet去训练

- 10^18次运算 = 90 epochs x 1.3M图片 x 7.7B (每张图片运算次数)
- 百亿亿次运算 (百亿的一亿倍)
- 1个低端M40 GPU: 实测需要14天
- 8个老P100 GPU: 实测需要29小时

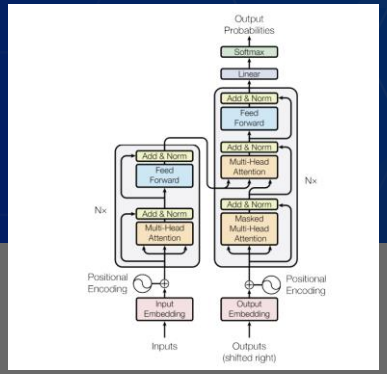
大模型算法技术分析——Encoder大模型

基础大模型结构为模型训练提供了基础架构

Colossal-AI

1 Attention

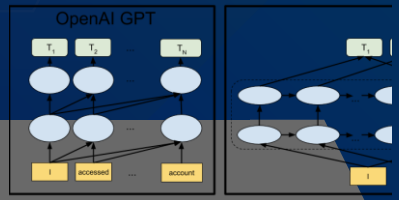
Google首创的Transformer大模型，是现在所有大模型最基础的架构，现在Transformer已经成为除了MLP、CNN、RNN以外第四种最重要的深度学习算法架构。



Transformer模型架构
(Vaswani, 2017)

2 BERT

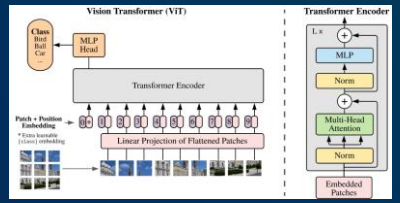
Google发布的首个预训练大模型BERT，从而引爆了预训练大模型的潮流和趋势。BERT强调了不再像以往一样采用传统的单向语言模型或者把两个单向语言模型进行浅层拼接的方法进行预训练，而是采用新的masked language model（MLM），以致能生成深度的双向语言表征。



BERT 模型架构
(Devlin, 2018)

3 ViT Google

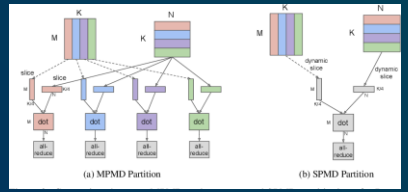
ViT Google提出的首个使用Transformer的视觉大模型。ViT作为视觉转换器的使用，而不是CNN或混合方法来执行图像任务。作者假设进一步的预训练可以提高性能，因为与其他现有技术模型相比，ViT具有相对可扩展性。



Vision Transformer
(Dosovitkiy, 2020)

4 Switch-Transformer

Google 将 Transformer 中的 Feedforward Network（FFN）层替换成了 MoE 层，并且将 MoE 层和数据并行巧妙地结合起来。在数据并行训练时，模型在训练集群中已经被复制了若干份。通过在多路数据并行中引入 All-to-All 通信来实现 MoE 的功能。



Switch 模型架构
(Fedus, 2021)

大模型算法技术分析——大模型盘点分析

Colossal-AI

具有里程碑意义性的大模型

1

GPT-3

OpenAI发布的首个百亿规模的大模型，应该非常具有开创性意义，现在的大模型都是对标GPT-3。GPT-3依旧延续自己的单向语言模型训练方式，只不过这次把模型尺寸增大到了1750亿，并且使用45TB数据进行训练。

2

T5: Text-To-Text Transfer Transformer

Google T5将所有 NLP 任务都转化成 Text-to-Text（文本到文本）任务。它最重要作用是给整个NLP预训练模型领域提供了一个通用框架，把所有任务都转化成一种形式

3

Swin Transformer:

微软亚研提出的Swin Transformer的新型视觉Transformer，它可以用作计算机视觉的通用backbone。在两个领域之间的差异，例如视觉实体尺度的巨大差异以及与文字中的单词相比，图像中像素的高分辨率，带来了使Transformer从语言适应视觉方面的挑战。

4

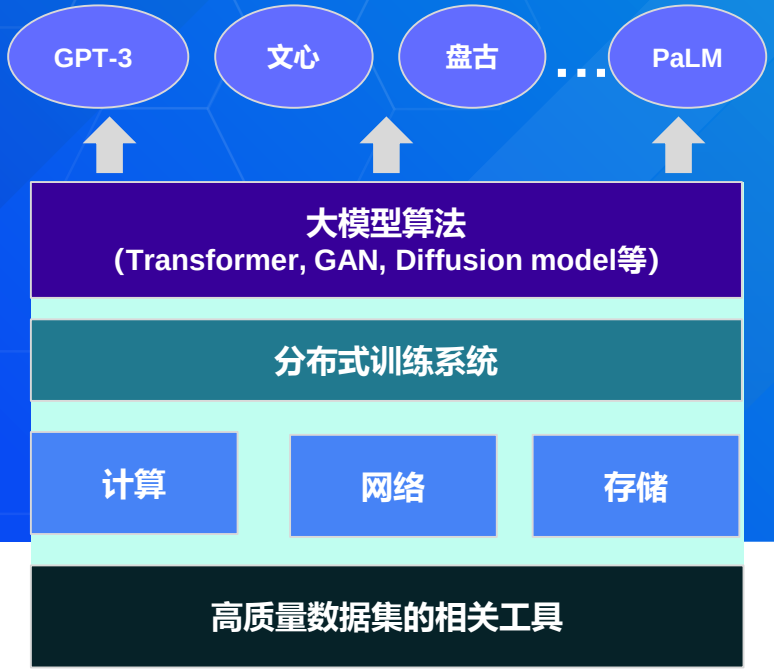
超过万亿规模的稀疏大模型

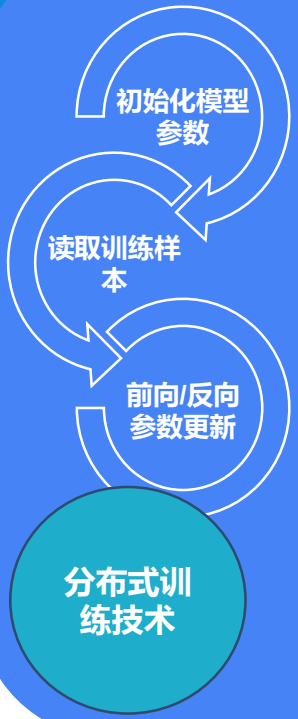
Switch Transformer：能够训练包含超过一万亿个参数的语言模型的技术。直接将参数量从GPT-3的1750亿拉高到1.6万亿，其速度是Google以前开发的语言模型T5-XXL的4倍。

超大规模基础模型训练核心技术

Colossal-AI

大规模基础模型训练涉及到大模型算法，分布式训练系统，优化方法，网络，和高质量数据集的相关工具等核心技术支撑





瓶颈：

单机单卡的作坊式训练已经无法满足模型训练的需要

显存占用大：
即便使用当前最大的GPU（NVIDIA最近发布的80GB-A100显卡），也无法在显存中拟合这些模型的参数

算力消耗大：
即使能够在单个GPU中拟合模型（例如，ZeRO-Offload通过在主机和设备存储器之间交换参数），所需的大量计算操作也会带来无法接受的训练时长（例如，使用单个V100 NVIDIA GPU，训练1750亿个参数的GPT-3将需要大约288年）。因此，分布式训练势在必行。

成本高昂：
高昂的硬件需求和训练成本仍严重阻碍着AIGC行业的快速发展。

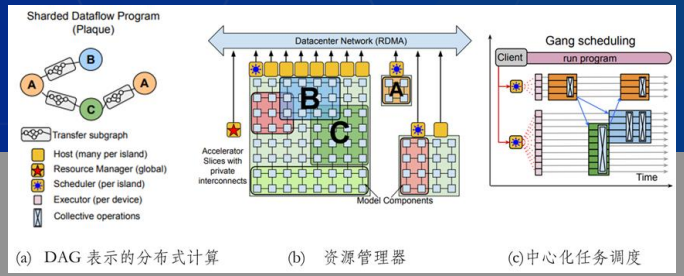
大模型算法技术分析——大模型盘点分析

Colossal-AI

具有里程碑意义性的大模型—— 比大更大：Pathways上实现的大语言模型PaLM

1 分布式框架Pathways

Pathways的很多重要思想来源于现有系统，包括用于表达和执行TPU计算的XLA、用于表征和执行分布式CPU计算的TensorFlow图和执行器、基于Python编程框架的JAX以及TensorFlow API。通过有效地使用这些模块，Pathways不需要对现有模型进行很多改动就能运行。



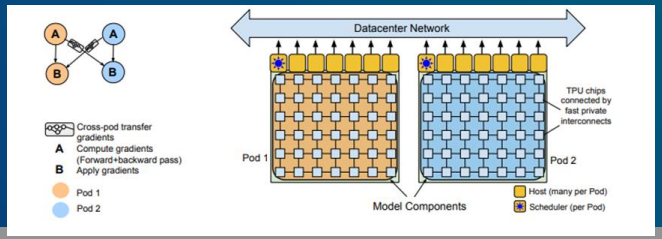
Pathways 框架
(Narang, 2022)

Pathways
分布式框架

PaLM模型

2 PaLM模型

PaLM吸引人眼球的是该模型具有5400亿参数以及采用新一代AI框架Pathways训练。模型结构也给出了很多方面优化，这些技术优化工作汲取了现有突出的研究成果，具体包括SwiGLU激活函数代替ReLU、层并行技术（Parallel Layers）、多查询注意力（Multi-Query Attention）、旋转位置编码（RoPE）、共享输入和输出词嵌入、去掉偏置参数（No Biases）等。

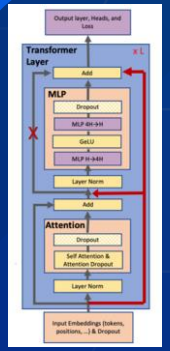


PaLM 模型架构
(Chowdhery, 2022)

大模型算法技术分析——大模型盘点分析

Colossal-AI

PaLM模型也是通过堆叠Transformer中的Decoder部分而成，该模型具有5400亿参数以及采用新一代AI框架Pathways训练。



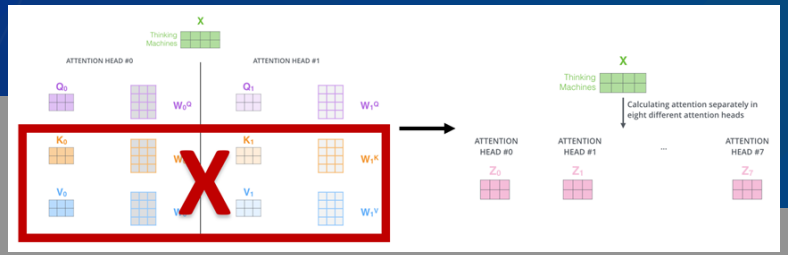
Transformer中的层并行
(Shoeybi, 2019)

1 层并行技术

PaLM从层间计算的并行角度考虑，通过将MLP和Attention共享输入实现MLP、Attention的并行计算，如图中粗红线标注。这种方法能在大规模训练中获得15%的提速，而且当模型达到62B之后没有性能损失。

2 共享K、V多注意力机制

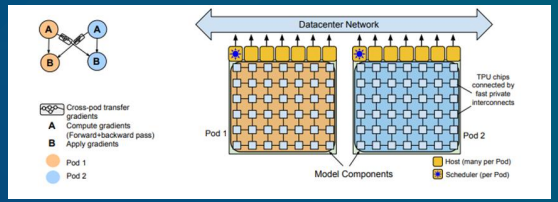
标准Transformer采用多注意力机制，每个时间点的输入张量与三个相同大小的权重矩阵相乘，得到同样大小的Q、K、V。PaLM保留了多头注意力机制，但对K、V在注意力头之间实现了参数共享。



注意力机制
(Vaswani, 2017)

3 训练

PaLM模型采用了两个TPU v4 Pod来完成540B参数的训练，每个Pod中含有3072个TPU v4芯片，整个模型共用了6144个芯片，两个Pod间通过DCN实现数据并行。



PaLM 模型架构(Chowdhery, 2022)

通过上述的训练方式，PaLM在新的BIG-bench基准上达到了平均人类水平，在0-shot、few-shot上也超过了GPT3。在具体下游任务上，PaLM在复杂逻辑推理任务上取得了突破性的语言熟练能力。

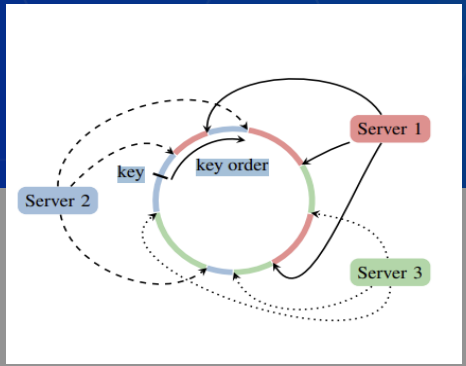
大规模分布式训练主要技术路线
——早期底层系统架构

Colossal-AI

过去几年中，神经网络规模不断扩大，而训练可能需要大量的数据和计算资源。模型底层架构的迭代在减少训练时间，保证高质量的计算能力上发挥着至关重要的作用。

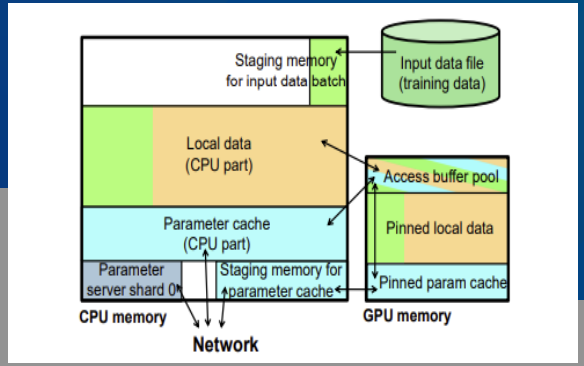
1 Parameter Server

工业界需要训练大型的机器学习模型，一些广泛使用的特定的模型在规模上的两个特点：1. 深度学习模型参数很大，超过单个机器的容纳能力有限；2. 训练数据巨大，需要分布式并行提速。OSDI版本偏向于系统设计，而NIPS版本偏向于算法层面。关于深度学习分布式训练架构来说是一个奠基性的存在。



2 GeePS 技术

分布式深度学习可以采用BSP和SSP两种模式。1为SSP通过允许faster worker使用staled参数，从而达到平衡计算和网络通信开销时间的效果。SSP每次迭代收敛变慢，但是每次迭代时间更短，在CPU集群上，SSP总体收敛速度比BSP更快，但是在GPU集群上训练，2为BSP总体收敛速度比SSP反而快很多。

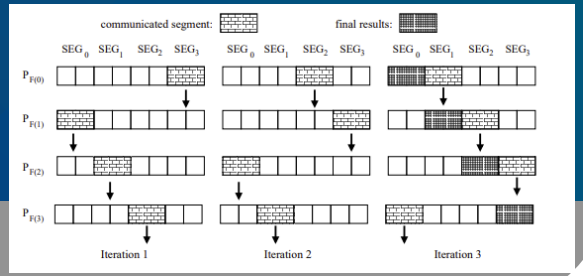


GeePS技术框架 (Cui, 2016)

3 HPC 技术

过去几年中，神经网络规模不断扩大，而训练可能需要大量的数据和计算资源。为了提供所需的计算能力，我们使用高性能计算（HPC）常用的技术将模型缩放到数十个GPU，但在深度学习中却没有充分使用。这种ring allreduce技术减少了在不同GPU之间进行通信所花费的时间，从而使他们可以将更多的时间花费在进行有用的计算上。

4 All-Reduce Algorithms



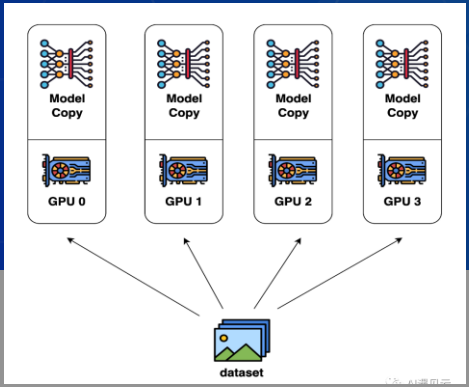
大规模分布式训练当前主要技术路线——并行训练技术

Colossal-AI

分布式训练并行技术即通过在训练过程中使用GPU集群（多机多卡）来提高神经网络的训练速度。

1 数据并行 (Data Parallelism, DP)

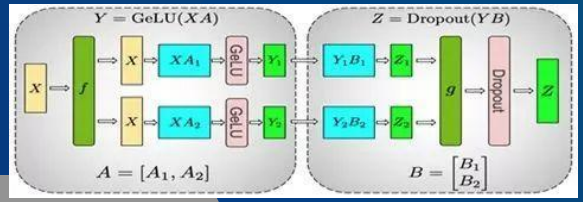
相同的设置和模型被复制多份，每份每次都被馈送不同的一份数据。处理是并行完成的，所有份在每个训练步结束时同步。



数据并行
(Juejin, 2023)

2 张量并行(Tensor Parallelism)

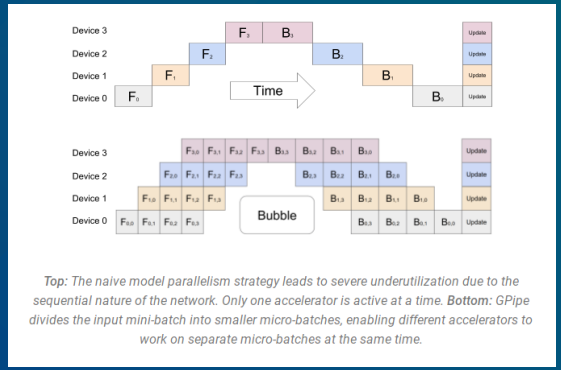
每个张量都被分成多个块，因此张量的每个分片都位于其指定的 GPU 上。在处理过程中，每个分片在不同的 GPU 上分别并行处理，结果在步骤结束时同步。



Megatron-LM 在计算
MLP 的并行过程

3 流水线并行(Pipeline Parallelism，PP)

模型在多个 GPU 上垂直 (即按层) 拆分，因此只有一个或多个模型层放置在单个 GPU 上。每个 GPU 并行处理流水线的不同阶段，并处理 batch 的一部分数据。



Top: The naive model parallelism strategy leads to severe underutilization due to the sequential nature of the network. Only one accelerator is active at a time. Bottom: GPipe divides the input mini-batch into smaller micro-batches, enabling different accelerators to work on separate micro-batches at the same time.

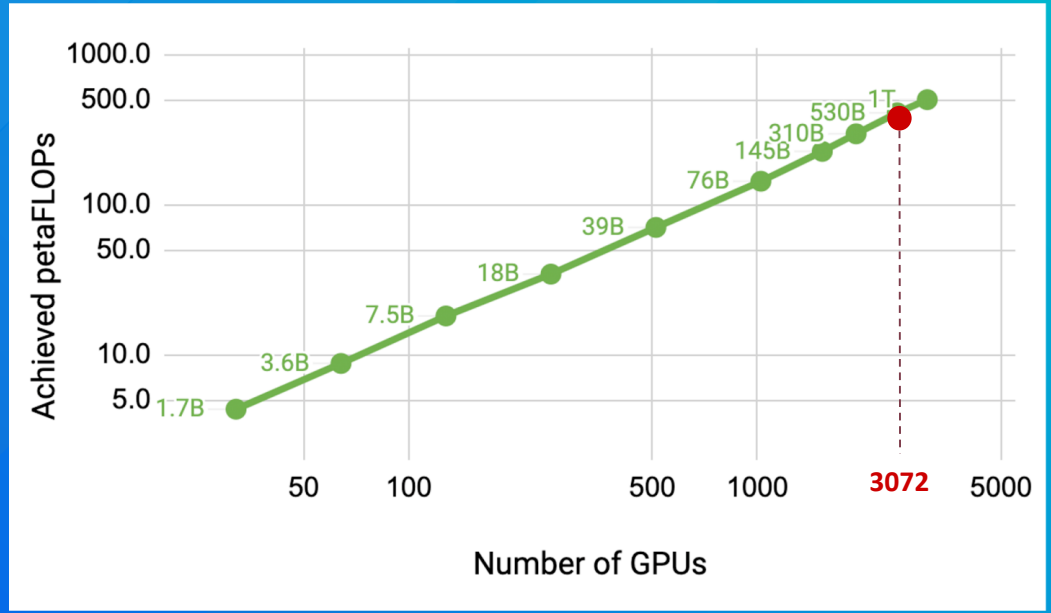
mp-pp

经典方案：英伟达3072个GPU

Colossal-AI

- 每个DGX服务器内的8个GPU用张量并行
- 64个服务器形成一个小组

  组内用流水线并行
- 6个小组之间用数据并行
- 8*64*6=3072 GPUs



https://developer.nvidia.com/blog/scaling-language-model-training-to-a-trillion-parameters-using-megatron/

大规模分布式训练主要技术路线——训练技术比较分析

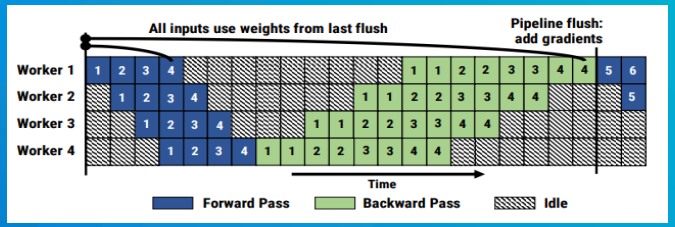
Colossal-AI

并行训练技术比较分析

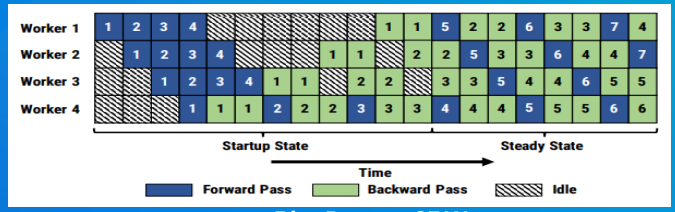
	数据并行	张量并行	流水线并行
通用性	完全通用	只能在部分模型上使用，比如CNN就暂时不能进行张量并行	基本通用
计算使用率	在数据分配较均匀的情况下，计算使用率损耗较小	需要经常进行中间结果的传输，计算使用率有一定损耗	流水线中存在气泡，计算使用率有一定损耗
显存开销	每张卡上都保留相同的模型和优化器参数，总开销比较大	能将显存开销均匀分摊到不同服务器上	如果模型不同层的参数量差异较大，则需要进行调整才能达到比较好的效果
通信量	只用传输不同GPU上的梯度，开销较小	需要经常的将中间结果进行传输，通信开销比较大	只用在不同层之间传输hidden state和反向梯度的值，通信开销较小
优势	通用性强且计算效率、通信效率较高	显存效率较高	显存效率较高，相比于张量并行的通信开销要小
劣势	显存总开销比较大	需要引入额外的通信开销，通用性不是特别好	流水线中存在气泡

流水线并行算法：GPipe 和 PipeDream

GPipe： 核心思想便是输入的minibatch划分成更小的micro-batch，让流水线依次处理多个 micro batch，达到填充流水线的目的，进而减少气泡。



PipeDream： 解决流水线气泡问题的方法则不一样，它采取了类似异步梯度更新的策略，即计算出当前 GPU 上模型权重的梯度后就立刻更新，无需等待整个梯度回传完毕。



PipeDream -2BW
(Narayanan, 2020)

为什么数据并行 (Data Parallelism)实用?

Colossal-AI

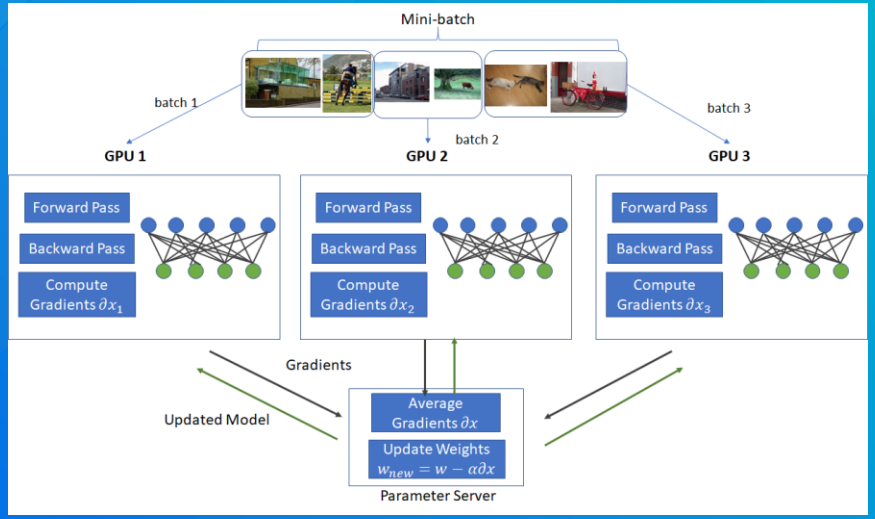
鲁棒性 (稳定性) 好

假定1000个机器只做数据并行

- 10个机器崩溃
- 结果基本不受影响
- 张量并行结果受影响
- 流水线并行结果受影响

算法清晰，易于实现

扩展性优化只需专注增大批量 (Batch Size)



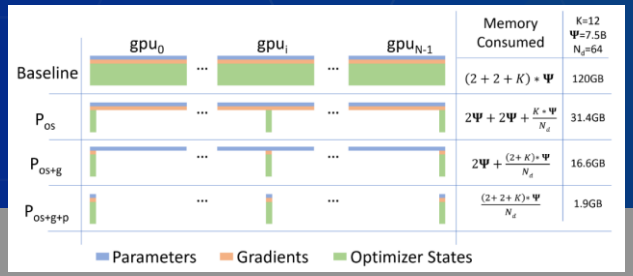
大规模分布式训练主要技术路线——显存优化技术

Colossal-AI

显存主要可以分为两大部分：常驻的模型及其优化器参数，和模型前向传播过程中的激活值。显存优化技术主要是通过减少数据冗余、以算代存和压缩数据表示等方法来降低上述两部分变量的显存使用量。

1 ZeRO 技术

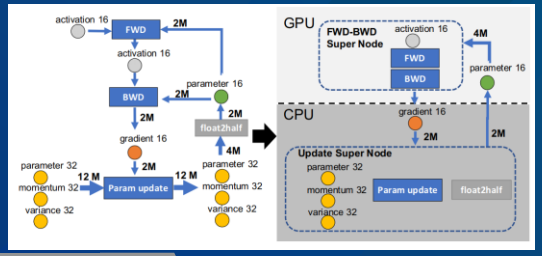
常驻在每块GPU上的数据可以分为三部分：模型参数，模型梯度和优化器参数ZeRO 技术便是分别考虑了上述三部分参数分开存储的情况。考虑每张卡上仅保留部分数据，其余的可以从其他 GPU 上获取。

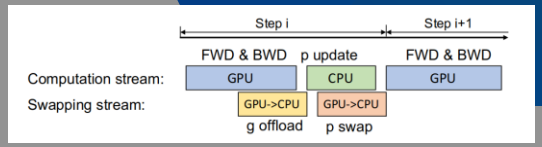


ZeRO 技术参数分开存储流程
(DeepSpeed, 2021)

2 Offload 技术

ZeRO-Offload [10] 技术主要思想是将部分训练阶段的模型状态 offload 到内存，让 CPU 参与部分计算任务。为了避免 GPU 和 CPU 之间的通信开销，以及 CPU 本身计算效率低于 GPU 这两个问题的影响。





CPU-offload 存储流程与参数传输流程
(DeepSpeed, 2021)

3 Checkpoint 技术

Checkpoint技术的核心是只保留checkpoint点的激活值，checkpoint点之间的激活值则在反向传播的时候重新通过前向进行计算。可以看出，这是一个以算代存的折中方法。

4 其他主流优化技术

名称	原理
大批量优化器	LARS/LAMB优化器解决使用大批量的训练方式可能导致模型训练不稳定问题
FP16	将原本的32位浮点数运算转为16位浮点数运算。一方面可以降低显存使用，一般还会配合动态放缩技术。
算子融合	将若干个 CUDA 上的运算合成一个运算，本质上是减少了 CUDA 上的显存读写次数。
自动并行	自适应的方法来设置超参数，被称为自动并行技术。这些方法包括动态规划、蒙特卡洛方法、强化学习等。

主流训练加速库汇总分析

Colossal-AI

名称	项目地址	STAR数量	计算框架	数据并行	模型并行	流水线并行	其他优化技术
Colossal-AI	https://github.com/hpcaitech/ColossalAI	28.1k	Pytorch	√	√ N-D Parallelism	√	ZeRO-DP, Adaptive-Offload, LAMB, LARS优化器
Horvod	https://github.com/horovod/horovod	13.2k	TensorFlow Pytorch	√			RING-AllReduce
Mesh-TensorFlow	https://github.com/tensorflow/mesh	1.4k	TensorFlow	√	√		
Megatron LM	https://github.com/NVIDIA/Megatron-LM	4.6k	Pytorch	√	√	√	
DeepSpeed	https://github.com/microsoft/DeepSpeed	18.9k	Pytorch	√		√	ZeRO-DP, ZeRO-Offload, LAMB, LARS优化器
BMTrain	https://github.com/OpenBMB/BMTrain	212	Pytorch	√			ZeRO-DP, ZeRO-Offload
LightSeq	https://github.com/bytedance/lightseq	2.7k	Pytorch TensorFlow	√			算子多运算融合
Alpa	https://github.com/alpa-projects/alpa	2.3k	Pytorch	√	√	√	支持自动并行
OneFlow	https://github.com/Oneflow-Inc/oneflow	4.9k	OneFlow	√	√	√	
FairScale	https://github.com/facebookresearch/fairscale	2.2k	Pytorch	√	√	√	ZeRO-DP

未来企业 AI 软件生态

Colossal-AI

Platform?

1

训练 + 微调 + 服务

10个模型 * 1万家企业 * 10个数据集 = 1百万份订单

巨大市场空间: 反复训练 & 服务

高效率

训练 & 推理

成本降低

2

APIs 影响数据隐私:

用户需要上传数据

未来商业模式: 出售新的模型(e.g., 软件授权)

打造高性能模型

ChatGPT

Stable Diffusion

DeepMind

HUGGING FACE

OpenAI

3

数据骤增:

数据集专注不同领域

数据处理

高效 & 数据清洗

scale

databricks

潞晨开源智能系统架构：Colossal-AI技术分析

Colossal-AI

底层硬件 → 潞晨解决方案 → 上层架构

CPU

GPU

TPU

FPGA

低延时推理系统

独创高效自动N维并行系统

异构内存管理系统

最小化部署成本　　最大化计算效率

PyTorch

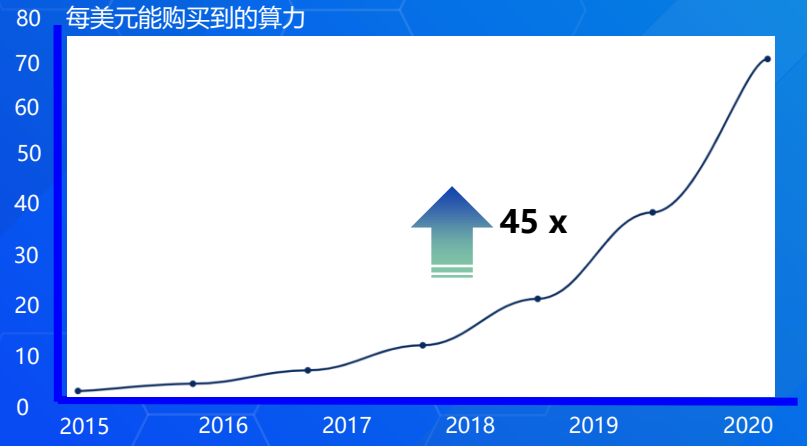
Keras

Hugging Face

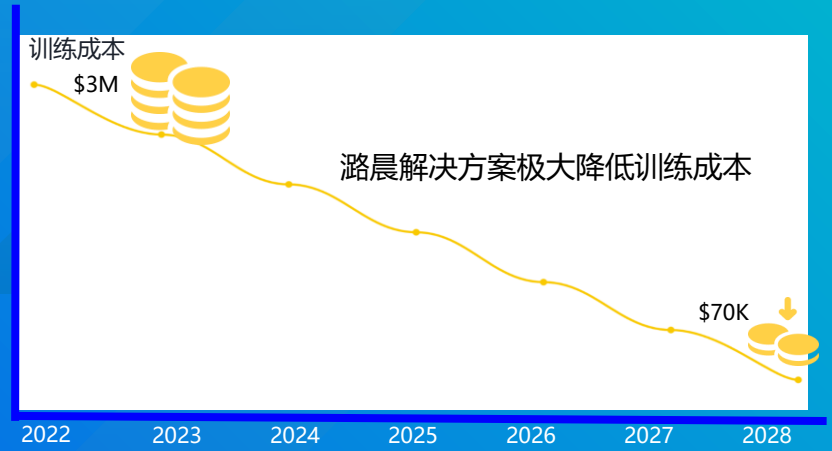
Lightning AI

愿景：百元微调ChatGPT

Colossal-AI



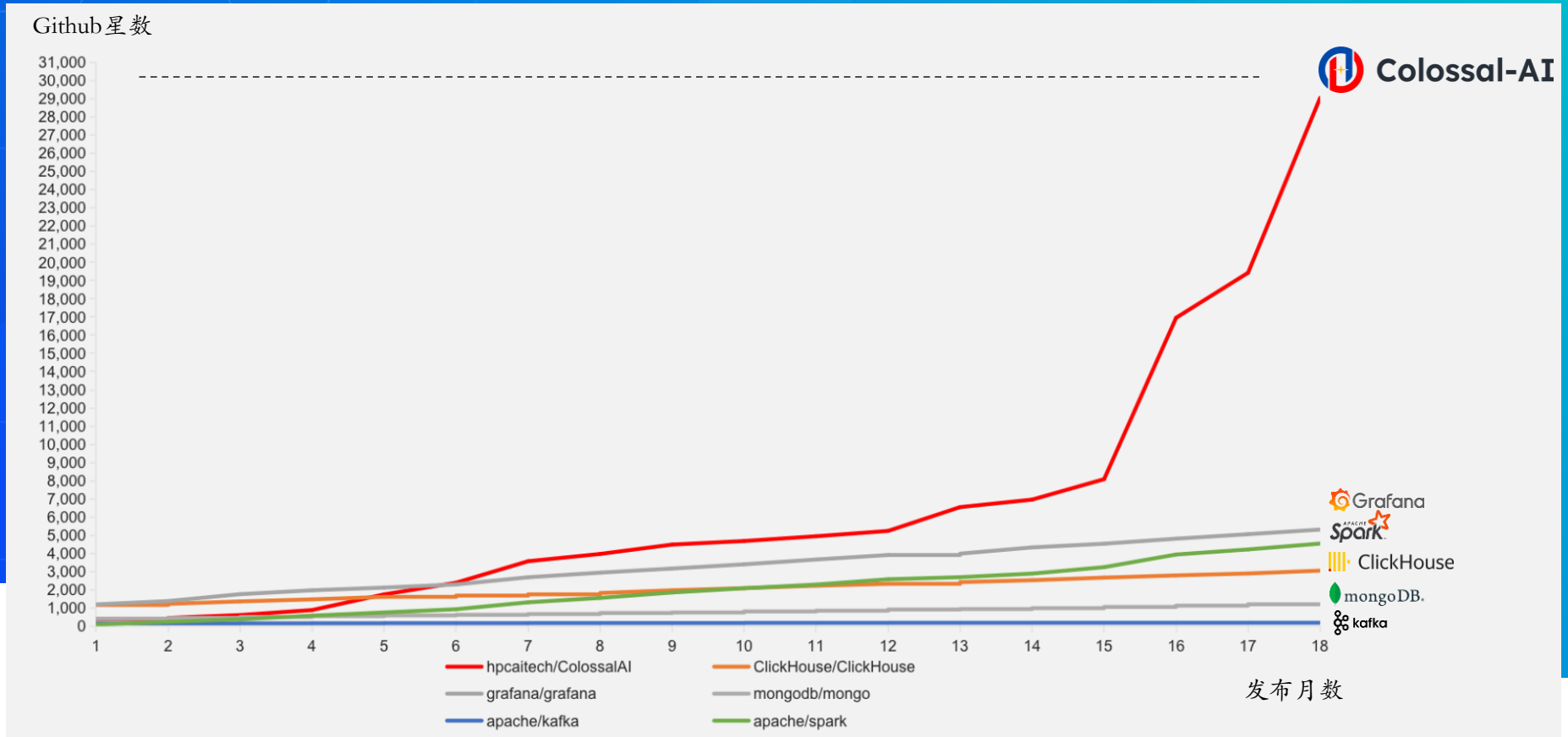
每美元买到的算力每年上升约1.86倍（AWS）



ChatGPT 训练费用有望降至 7 万美元

社区发展速度远超主流开源项目

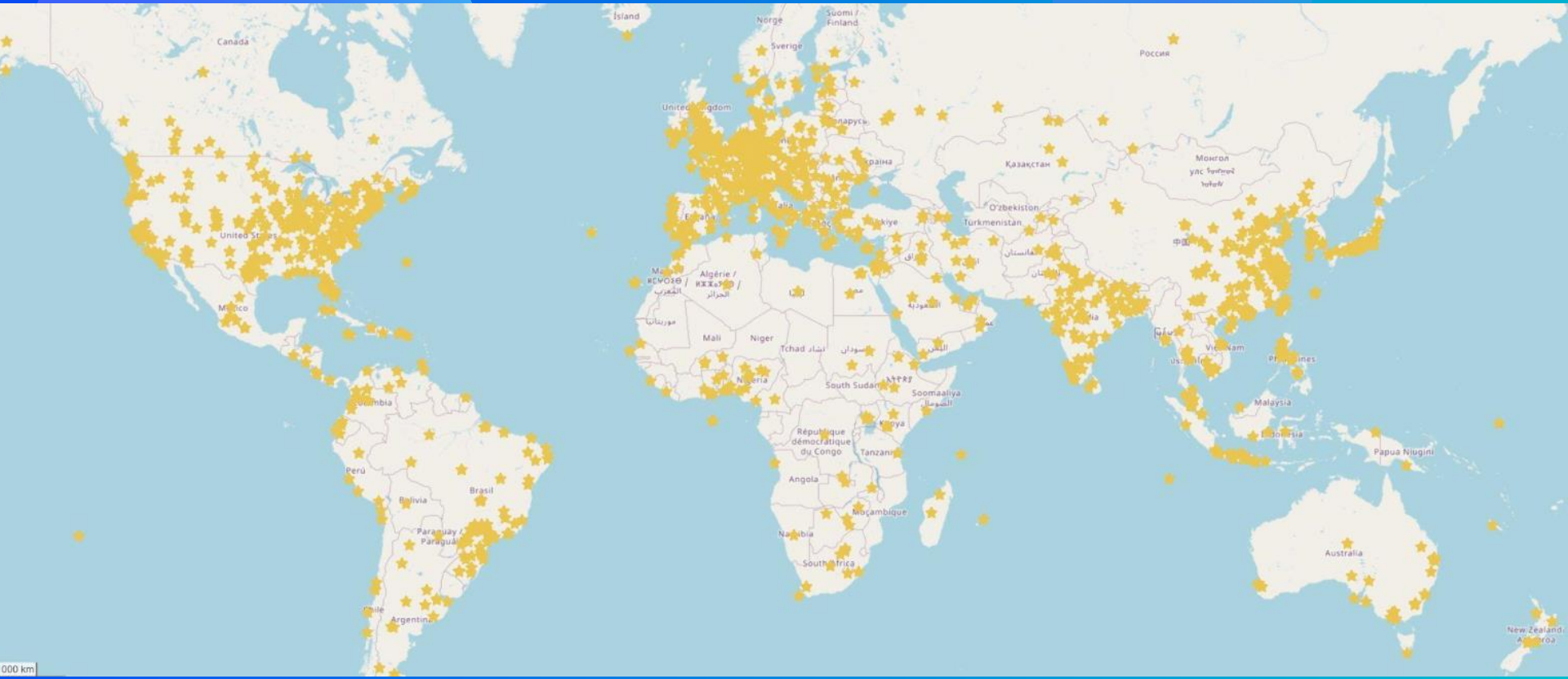
Colossal-AI



https://github.com/hpcaitech/ColossalAI

Colossal-AI 用户遍布全球

Colossal-AI



https://github.com/hpcaitech/ColossalAI

Visualized by: https://github.com/python-visualization/folium


THANKS

主要参考文献

[1] “A timeline of large language models”, Reddit, 2023
[2] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, etc, “PaLM language modeling with Pathways”, 2022, Google Resarch, arXiv: 2204.02311v5 [cs.CL] 5 Oct 2022.
[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser and L. Polosukhin, “Attention is all you need”, arXiv:1706.03762
[4] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. Liu, K. Malkan, N. Fiedel and M. Dinculescu. “Exploring Transfer Learning with T5: the Text-To-Text Transfer Transformer”, Google Research, https://ai.googleblog.com/2020/02/exploring-transfer-learning-with-t5.html
[5] A. Dosocitskiy, L. Beyer, A. Kolesnikov,D. Weissenborn, etc. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, 2020, arXiv:2010.11929
[6] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.
[7] H. Cui, H. Zhang, G. Ganger, P. Gibbons, and E. Xing. “GeePS: Scalable Deep Learning on Distributed GPUs with a GPU-Specialized Parameter Server. In ACM European Conference on Computer Systems”, 2016 (EuroSys'16)
[8] J. Yang, G. Xiao, S. Yulong, W. Jiang, X. Hu, Y. Zhang and J. Peng, “A survey of knowledge enhanced pre-trained models”, arXiv:2110.00269v1 [cs.CL] 1 Oct 2021
[9] N. Heidloff, “Foundation Models, Transformers, BERT and GPT”, https://heidloff.net/article/foundation-models-transformers-bert-and-gpt/
[10] R. Horev, “BERT Explained: State of the art language model for NLP”, Medium, https://heidloff.net/article/foundation-models-transformers-bert-and-gpt/
[11] Shoeybi, Mohammad, et al. "Megatron-lm: Training multi-billion parameter language models using model parallelism."
[12] S. Narang and A. Chowdhery, “Pathways Language Model (PaLM) scaling to 540 billion parameters for breakthrough performance”, 2022, Google Research, https://ai.googleblog.com/2022/04/pathways-language-model-palm-scaling-to.html
[13] W. Fedus, B. Zoph and N. Shazeer, “Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity”, 2021, arXiv:2101.03961
[14] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. 31st Conference on Neural Information Processing Systems 2017(NIPS 2017). Long Beach, CA, USA: 2017: 5998–6008.
[15] X. Han, Z. Zhang, N. Ding, Y. Gu, X. Liu, Y. Huo, J. Qiu, Y. Yao, A. Zhang, L. Zhang, W. Han, M. Huang, Q. Jin, Y. Lan, Y. Liu, Z. Liu, Z. Lu, X. Qiu, R. Song, J. Tang, [16] J. Wen, J. Yuan, W. Zhao and J. Zhu, “Pre-trained models: Past, present and future”, AI Open, Volume 2, 2021, https://doi.org/10.1016/j.aiopen.2021.08.002.
[16] Y. Wang, J. Kwok, L. M. Ni and Q. Yao, "Generalizing from a few examples: A survey on few-shot learning", arXiv:1904.05046, 2019