Cloudless-Training：提升地理分布式机器学习训练效率的框架

Wenting Tan[1,2], Xiao Shi[1, *], Cunchi Lv[1,2], Xiaofang Zhao[1]

[1] 中国科学院计算技术研究所，北京，中国[2]中国科学院大学，北京，中国
{tanwenting, shixiao, lvcunchi21s, zhaoxf}@ict.ac.cn

摘要——**地理分布式机器学习训练能够利用多区域云资源和广域网，支持许多新兴的机器学习场景（例如大模型训练、联邦学习）。然而，其效率受两项挑战限制。首先，通常缺少对多区域云资源进行高效弹性调度的能力，影响训练的资源利用率和性能。其次，广域网上的训练通信仍是主要开销，且容易受到广域网低带宽和剧烈波动的影响。**

本文提出了一个框架 Cloudless-Training，从三个方面实现高效的基于参数服务器的地理分布式机器学习训练。首先，该框架采用包含控制平面和物理训练平面的两层架构，以无服务器方式支持多区域云的弹性调度和通信。其次，它提供了一种弹性调度策略，可根据可用云资源的异构性和现有训练数据集的分布情况，自适应地部署训练工作流。第三，它为云之间的训练分区提供了两种新的同步策略，包括带梯度累积的异步随机梯度下降（ASGD-GA）和参数服务器间模型平均（MA）。该框架基于 OpenFaaS 实现，并在腾讯云上进行了评估。实验表明，Cloudless-Training 能够以地理分布式方式支持通用机器学习训练，大幅提升资源利用率（例如降低 9.2%–24.0% 的训练成本）和同步效率（例如相较基线最高实现 1.7 倍训练加速），同时保证模型正确性。

关键词——地理分布式机器学习训练、无服务器、弹性调度、同步

I. 引言

地理分布式机器学习训练可以通过多区域灵活配置云资源和分区训练数据集，满足大规模训练或因业务需求、隐私原则而开展的协同训练要求。例如，单个云中实时可用的资源可能无法满足大模型训练的需求；多模态训练数据集（如用户行为日志、图像、视频）则可能以极高的速率生成，并存储在世界各地的云中。所有这些因素都推动了在多区域云上开展地理分布式机器学习训练的必要性，如图 1 所示。目前，这种训练已应用于许多新兴的机器学习场景，例如大模型训练[1]，联邦学习[2], 以及协同边缘云[3] 或混合云[4] 环境中的机器学习训练。

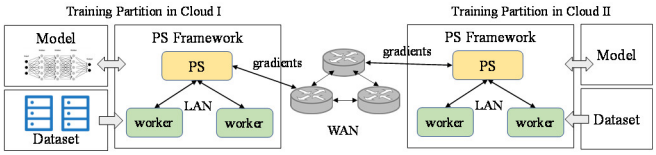

图 1. 地理分布式机器学习训练方法的架构 d

然而，由于复杂环境中的负载均衡和通信管理面临挑战，用户要实现高效的地理分布式训练并非易事。首先，通常缺少对多区域云资源进行高效弹性调度的机制，导致负载不均衡。这会直接降低资源利用率、训练速度和性能。例如，在腾讯云的 2 个云区域中使用相同数量的 CPU 核心训练 LeNet[43]，若数据分布不均，会导致其中 1 个区域的资源超额配置 25%，如图 2 所示。其次，广域网上的模型同步通信仍是主要开销，且容易受到广域网低带宽和大幅波动的影响。为保持各云之间的模型状态一致，需要定期通过通信交换各训练分区的模型副本。采用参数服务器（PS）模式[12],时，每个云级参数服务器都需要与全局模型副本通信并对其进行更新；通信开销会随模型规模和网络带宽而变化，这给云间广域网带来了很大压力。例如，在腾讯云的 2 个云区域中通过 GPU 训练 ResNet18[6]，使用 100Mbps 广域网时，广域网通信时间占总时间的 98% 以上，如图 3 所示。

已有研究探索了对各种地理分布式机器学习场景的支持，包括训练和推理[8][9][2][3][4]。Gaia 将云内通信与云间通信解耦，并提出了一种新的机器学习同步模型——近似同步并行（ASP），以动态消除云间不重要的通信[8]。Nebula-I 采用混合并行计算方法、自适应通信加速技术和数据压缩方法[9]。这些研究提出了提高广域网同步效率的方法，但未解决上述负载均衡挑战。在通信优化方面，这些模型采用启发式方法，仍有进一步研究的空间。

本文提出 Cloudless-Training，这是一个从三个方面提升地理分布式机器学习训练的资源配置与同步效率的框架。首先，它采用两层架构，借助无服务器范式将控制与实际训练解耦，通过多区域弹性调度和同步支持简化训练管理的逻辑视图。其次，它提供一种弹性调度策略，根据可用云资源的异构性以及各云中已有训练数据集的分布情况，部署资源分配适当的训练工作流。第三，它为云之间的训练分区引入了两种模型同步策略，包括带梯度累积的异步随机梯度下降（ASGD-GA）和模型平均（MA）。这两种策略都是经过调整的变体，旨在满足地理分布式环境的需求。该框架基于

OpenFaaS[7] 实现，并在腾讯云的两个区域（上海和重庆）通过 100Mbps 广域网进行了评估。

总而言之，我们的主要贡献有以下四点：

1. 我们设计了两层基于参数服务器的地理分布式机器学习训练框架，并采用无服务器范式，大幅提升了训练过程中的资源利用率和同步效率。

2. 我们提出了一种弹性调度策略，通过对可用云资源的异构性以及现有训练数据集的分布进行建模，确保实现负载均衡的资源供给。

3. 我们采用两种策略，在训练分区之间进行多区域模型同步，分别是带梯度累积的异步随机梯度下降（ASGD-GA）和模型平均（MA）。这两种策略都能显著降低广域网带来的开销。

4. 我们在真实的多区域云环境中实现并评估了 Cloudless-Training，结果表明，地理分布式训练的资源利用率和训练性能均可大幅提升。该框架在保证模型正确性的同时，可提高资源利用率（例如，降低 9.2%–24.0% 的训练成本）和同步效率（例如，训练速度最高可达到基线的 1.7 倍）。

II. 背景与挑战

A. 地理分布式机器学习训练

与简单的机器学习场景相比，许多新兴的机器学习应用带来了新的需求，并显著推动了地理分布式机器学习训练的实践，以实现高性能、大规模或协同训练。在此，我们总结了两项基本需求。

需求 1. 灵活获取比单个云更多的云资源。 随着模型规模和数据集容量不断增长，单个云中实时可用的资源可能无法满足大型模型的训练需求。人工智能应用不断要求数量更多、质量更高的计算资源[17][18]。例如，英伟达开发的 Megatron LM 模型拥有 83 亿个参数，使用 512 个英伟达 Tesla V100，在 174 GB 的文本语料上进行训练[29]。因此，小规模云数据中心有时难以满足其需求。将训练任务拆分到多个云中，自然可以从其他云获取补充资源。基于这一策略，地理分布式机器学习训练可用于支持在各种模型以及边缘云[19]，、混合云[4] 或多云环境中的训练。

需求 2：利用分布在多区域云中的数据进行训练。 出于延迟或服务质量方面的考虑，许多应用分布在多区域云中。数据通常存储在本地，对于构建人工智能助手服务十分重要。在这种情况下，如何实现机器学习训练需要经过审慎考量。数据（例如用户行为日志、图像、视频）可以在全球各地以极高的速率生成 [20][21]。例如，TikTok 用户通过观看视频产生海量信息，这些视频由其全球部署的云服务提供[22]。汇集所有数据面临实际挑战，例如广域网带宽低、传输成本高，以及严格的数据隐私规则；联邦学习[2] 因此，地理分布式机器学习训练可以用于协调多个云之间的训练。

目标。 为满足这些要求，构建一个高效的地理分布式机器学习训练框架颇具吸引力。该框架应具备可扩展性，并能为后续实践与研究提供启发。常见的并行训练模式包括数据并行 [14], 模型并行[15] 和流水线并行[16]。在这项工作中，我们重点改进采用参数服务器架构[12], 的数据并行训练，该架构是支持数据并行的经典架构。采用参数服务器进行地理分布式机器学习训练的效率仍然有限。尽管已有研究提出了许多解决方案[8][9], , 但这些研究构建的框架并未考虑日益兴起的云计算趋势，例如无服务器计算[24], , 而无服务器计算有助于减轻训练过程中的管理负担。在这项工作中，我们认为，无服务器计算凭借云函数细粒度扩展这一固有优势，能够简化框架构建，并更好地满足这些要求。此外，我们发现，负载不均衡和通信开销是导致地理分布式机器学习训练实践面临显著挑战的两个主要因素。

B. 挑战 *1*：跨云弹性调度

在地理分布式机器学习训练中，各云之间很容易出现负载不均衡，因此需要高效的弹性调度。

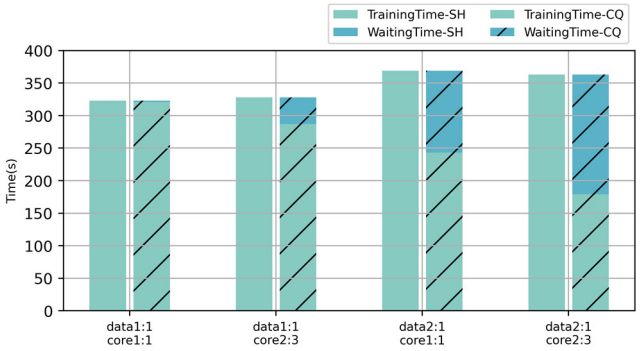

图 2. 在腾讯云上海和重庆区域，采用不同类型的异构资源分配和不均匀数据分布训练 LeNet 的时间占比

多区域云之间的负载不均衡。 通常，针对各云中的训练任务制定资源配置计划时，依赖经验或采用粗略的方法（例如贪心策略）。考虑到多区域云中的异构资源和不均匀数据分布，要找到合适的配置以避免负载不均衡并不容易。这会显著降低资源利用率、训练速度和训练性能。如图 2 所示，随着数据分布比例以及可用计算资源的类型和数量发生变化，就会产生负载不均衡，而训练速度完全取决于负载最大的节点。同时，负载较轻的节点会占用计算资源，等待掉队节点赶上，从而造成不必要的资源消耗。此外，这会加剧各云中的训练异步性，并显著降低性能，因为节点可能会从掉队者处收到严重陈旧的梯度或参数。例如，AdamLike 模型[25] 表明，梯度陈旧性会严重阻碍收敛过程。

目标。 为了提高地理分布式机器学习训练中的资源利用率、速度和性能，有必要

平衡各云之间的相对负载，使每个云中的训练能够以更协调的方式进行处理。然而，以往的研究通常假设训练是在各云之间资源同构、事件数据分布均衡的情况下进行负载均衡的，例如 Gaia[8], Nebula-I[9]。为了使负载均衡调度能够动态适配每次训练，我们构建了一个模型，用于量化可用资源的负载状况和数据分布。此外，我们提倡采用无服务器方式，通过按需分配和回收训练工作节点来提高资源伸缩效率，减少因等待拖尾节点而产生的长时间等待。

C. 挑战*2*：降低广域网上的同步开销

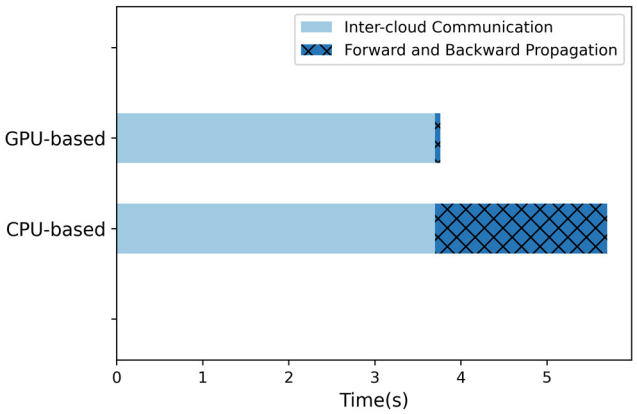

图3：在腾讯云的上海和重庆两个区域，使用中央处理器或 GPU 训练 ResNet18 的时间占比

在地理分布式机器学习训练中，通过广域网同步模型的中间结果所产生的开销，远高于通过局域网同步的开销，并会在运行期间周期性地影响（例如暂停或延迟）训练过程。这种开销不容忽视。

广域网上的同步开销。在机器学习训练中，每个工作节点都需要频繁地与参数服务器或其他工作器通信，以保持模型更新。然而，在地理分布式机器学习训练中，部分通信会转变为通过广域网进行的跨云模式，而广域网通常只能提供数百 Mbps（例如 100 Mbps）的带宽，至少比云内局域网慢 50 倍[23]。这会提高通信在训练中的占比。如图3所示，在腾讯云的重庆和上海区域训练 ResNet18[6]（模型规模为 48 MB）时，使用中央处理器和 GPU 训练的广域网通信时间分别占总训练时间的 64.9% 以上和 98.4%。这不仅会减慢训练过程，还可能因广域网流量收费，或因等待同步而间接浪费资源，导致用户承担更多成本。

目标。降低广域网上的同步开销，有助于提升训练速度和性能，并降低成本。在优化简单机器学习训练中的同步方面，现有研究主要采用两种方法：压缩同步数据量，例如 DGC[13], top-K[35]，；或降低同步频率，例如梯度累积[26], 和模型平均[27]。梯度累积策略会累积本地梯度，直到满足同步条件或达到同步阈值。模型平均策略则定期对并行工作节点各自训练的模型进行平均。Gaia[8] 等工作采用降低同步频率的方法，将云内通信与云间通信解耦，并使用近似同步

并行（ASP）以消除不显著的通信。然而，由于微小的参数更新可能会影响模型收敛，因此很难准确定义模型更新的显著性阈值。本文首先考虑通信寻址机制，以支持 Cloudless-Training 无服务器架构中的云函数通信。另一方面，为降低同步频率，我们采用了两种前述方法的变体：带梯度累积的异步随机梯度下降（ASGD-GA）和参数服务器间模型平均（MA）。据我们所知，这是首次将它们用于基于参数服务器的地理分布式机器学习训练。ASGD-GA 为训练分区提供了一种异步协调策略，而以往工作中的训练分区通常采用同步方式；它还结合梯度累积来减少信息损失。MA 则是一种面向模型参数的方法。

I. CLOUDLESS-TRAINING 设计

A. 框架概述

架构。Cloudless-Training 为实现基于参数服务器的地理分布式机器学习训练提供了整体视图。它采用两层结构，包括控制平面和物理训练平面。它提供了一种逻辑视图：来自不同云的所有训练分区都汇集到一个统一的逻辑平面中，如图4所示。与传统的地理分布式机器学习训练框架相比，Cloudless-Training 采用无服务器风格构建。在控制平面中，主要负责调度训练任务，并支持所有云成员之间的通信寻址，使每个云级训练分区都能并行运行，并获取对等方的参数服务器通信器云函数的全局通信地址。控制平面可通过无服务器方式部署在任意云中，用户可通过提供模型定义和训练配置，经由该平面提交训练任务。在物理训练平面中，每个云中的训练形成一个分区，并以无服务器工作流的形式进行组织、部署和运行。在内部，该工作流实现常规机器学习训练的功能；在外部，它通过通信器函数定期与其他工作流通信，并更新本地模型状态以供本地使用。除训练启动阶段外，每个云中的所有训练分区均以去中心化方式运行。

训练工作流调整。根据架构，完整的训练工作流跨多个云运行。与简单的训练无服务器工作流相比，Cloudless-Training 中的训练工作流进行了两项主要调整。首先，在控制平面中加入云函数以进行协调，包括调度函数和通信器寻址函数。它们在启动阶段运行。其次，修改参数服务器云函数的功能，以支持广域网同步。这样，每个云中的经过修改的参数服务器函数就可以通过通信器函数在广域网上公开其身份，并与其他参数服务器函数管理中间状态。

该工作流以无服务器方式扩展，以服务一项训练。收到训练请求后，调度器函数首先响应，加载调度策略，为每个云生成训练计划，并调用子

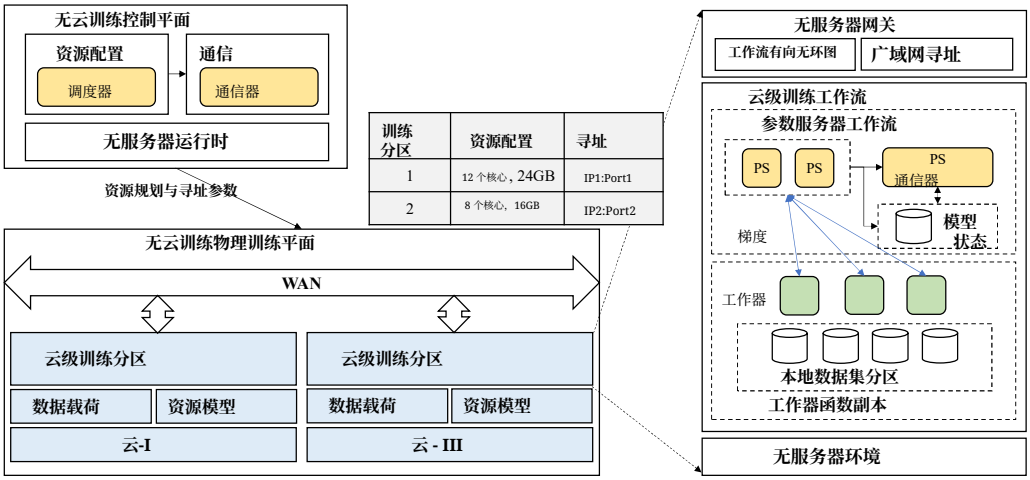

图4. Cloudless-Training 的架构

工作流在每个云中运行。随后，全局通信器函数等待每个云中的参数服务器函数就绪，并为每个参数服务器通信器分配通信地址，将其无服务器身份映射到广域网上的 <IP、端口> 。

准备工作完成后，每个云级训练分区都会部署相应的训练函数并在本地执行，包括访问本地训练数据、计算本地随机梯度以及更新本地模型描述。在预定的同步时间点（例如，每 10 次迭代），每个参数服务器通信器都会将本地模型状态传输到另一个参数服务器。参数服务器收到远程模型状态后，会据此更新本地状态，并让本地工作器随后访问新版本。为降低资源消耗，每个云中的工作函数会在本地训练完成后立即终止（例如，当轮数或损失在本地达到阈值时）。

调度机制。该调度机制支持集中式调度策略，并可在必要时通过更新调度器云函数的实现来扩展，以支持更多策略。在这项工作中，我们提出了一种弹性调度策略，以实现云之间负载均衡的资源供给，详见 B 节。该机制允许探测或手动设置可用云资源和数据分布状态，以评估云异构性，并根据弹性调度策略模型生成资源规划。

同步支持。在简单的机器学习训练中，由于云函数具有弹性，且参数服务器云函数是有状态的，因此每个参数服务器在其本地无服务器运行时中都有一个地址，可以由本地函数稳定访问。而对于广域网通信，全局通信器函数需要为每个参数服务器通信器分配唯一身份，以便进行广域网通信。为减少广域网上的通信流量，Cloudless-Training 限制每个参数服务器每次只能向另一个参数服务器发送其状态。因此，通信器需要规划通信拓扑，并在准备阶段或发生重新调度时通知每个参数服务器。此外，它还提供两种策略，在保证训练正确性的前提下，减少多个云之间的同步频率，包括带梯度累积的异步随机梯度下降和参数服务器间模型平均方法，详见 C 节。

B. 弹性调度策略

弹性调度策略的灵感源自不同训练阶段的时间构成。它量化相对计算负载，以平衡关键计算阶段的耗时，进而形成一种启发式调度算法。

调度洞察与负载建模。 以广域网上每次的同步时刻为界，这些时间间隔可视为每个训练分区本地处理的一个重复周期（$T_{process}$）。在此间隔内，每个云级训练都会完成模型加载（T_{load}）、包含前向传播与反向传播的本地训练（T_{train}）以及本地通信，并等待其他云中的对等方赶上进度或完成训练。由于训练过程是有状态的，且训练期间云资源不会释放，因此等待时间会造成不必要的资源消耗。这种关系可表示为 $process = _{load} + train$。如 II.B 节所述，如果各个云的 $T_{process}$ 差异较大，则无法保证训练性能。根据研究 [11] 和常识，T_{train} 是$T_{process}$的主要组成部分，并且会直接受到负载均衡状态的影响。

表 I. 云资源训练速度量化 s

CPU/GPU	*Used* 核心数	每秒万亿次浮点运算/ *TN*	迭代时间（秒）/*IN*	*IN/TN* 比率
英特尔至强 IceLake（基线）	2	0.096/ 1.000	3.697/ 1.000	1.000
英特尔至强 Cascade Lake	2	0.090/ 0.938	5.549/ 0.666	0.710
英特尔至强 Skylake	2	0.112/ 1.167	3.800/ 0.973	0.834
英伟达 T4	2560	5.554/ 57.854	0.062/ 59.629	1.031
Nvidia V100	5120	13.345/ 139.010	0.024/ 154.042	1.108

通过考察分配的云资源与数据分布是否匹配，可以大致预测负载均衡状态。我们从云中的部分计算设备（包括中央处理器和 GPU）采集其规格信息（例如中央处理器的类型、核心数或 CUDA、每秒万亿次浮点运算），并通过观察实际性能（例如使用

数据集 cifar-10[28], 训练 ResNet18 的迭代时间）来量化其计算能力，并使用基线对指标进行归一化（例如，TFLOPS归一化（TN）、迭代时间归一化（IN）以及 IN/TN 的比值），以便在训练过程中进行比较，如表 I 所示。显然，在每次训练迭代中，T_{train} （以迭代时间量化）与设备的计算能力（C_{device}，以 TN 量化）成反比。总体而言，T_{train} 也与数据集的大小（S_{data}）成正比。因此，我们得到 $train \propto S_{data} C_{device}$。对于每台设备，都可以通过其 TFLOPS 值或经验迭代时间大致预测或比较其 C_{device} 。

接下来，可以定义特定云 i 的训练负载功率（LP）。它描述了使用特定资源处理特定负载的能力。粗略而言，可将其量化为可用设备（如中央处理器、GPU）的计算能力总和与本地数据集大小之比，如式（1）所示。这里，m、n 分别表示可分配用于训练的中央处理器和 GPU 设备数量，这些设备可能由用户预留，也可能由用户实时申请。该指标可用于预测采用不同资源分配方案时训练的相对负载状态，是调度算法中的重要指标。

$$LP_i = \frac{\sum_{m=1}^{M} N_{cpu,m} * P_m + \sum_{n=1}^{N} N_{gpu,n} * P_n}{S_{data}} \quad (1)$$

调度算法。 此外，我们提出一种算法，为每个云规划训练资源配置，以实现负载均衡的弹性调度，如表 II 所示。其核心思想是比较每个云的负载功率单位（LP_i），找出最小的负载单位（它可能是最严重的拖尾节点），并将其作为参照，通过穷举搜索确定每个云所需的实际资源数量，以匹配该拖尾节点（search_optimal_plan()）。

表 II. 弹性调度算法

算法 1 最优匹配算法

输入:
N: 云的数量,
Res[N]: 所有分区的计算资源,
$S_{data}[N]$: 所有分区的数据集大小。
输出:
ResPlan[N]: 每个云的资源规划。

LP[N] = {}
MinLP = $+\infty$
ResPlan[N] = {}
对于 *i*, *res* **属于** *Res[N]* **执行**
    LP[i] ← 获取_*LP(res)* // 公式 *(1)*
    最小线性规划 ← **最小值**(最小线性规划, 线性规划*[i]*)
结束 对于
对于 *i*, *res* **属于** *Res[N]* **执行**
    ResPlan[i] ←搜索_最优_*plan(res)*
结束循环
返回 *ResPlan[N]*

C. 广域网上的同步优化Cloudless-Training 基于广域网，在工作流之间构建了一种基本的广域网同步机制提供通信支持，并提供两种策略：带梯度累积的异步随机梯度下降（ASGD-GA）和 PS 间模型平均方法（MA）。它们分别通过降低同步频率，在不损害模型正确性的前提下支持梯度和模型参数的同步。ASGD-GA 为地理分布式训练分区提供一种异步协调策略，而这些分区以往通常采用同步方式；此外，它还结合梯度累积以减少信息损失。相比之下，MA 主要面向模型参数。因此，这两种策略分别代表基于梯度和基于参数的模型同步方法。

基本的广域网同步机制以周期性方式，在不同云之间的发送方和接收方参数服务器通信器之间同步模型状态，具体如下:

1. 在每次训练迭代中，每个训练分区的工作函数从本地参数服务器拉取最新模型，执行随机梯度下降，并将梯度推送到本地参数服务器。2. 本地参数服务器使用随机梯度下降更新本地模型状态。本地更新过程可以是同步或异步的。3. 本地参数服务器检查同步策略（ASGD-GA 或 MA）的同步条件，以决定此时是否需要与其他云进行模型同步。如果不需要，本地参数服务器便结束该过程。4. 本地参数服务器通信器打包待发送状态，并根据同步策略（ASGD-GA 或 MA）的设置与接收方参数服务器通信。5. 接收方参数服务器根据同步策略的算法更新其本地模型状态。

具体而言，ASGD-GA 和 MA 在同步条件、待发送状态（梯度或模型参数）、通信模式（所有训练分区之间同步或异步）以及更新算法（随机梯度下降或模型平均）的设置上有所不同。

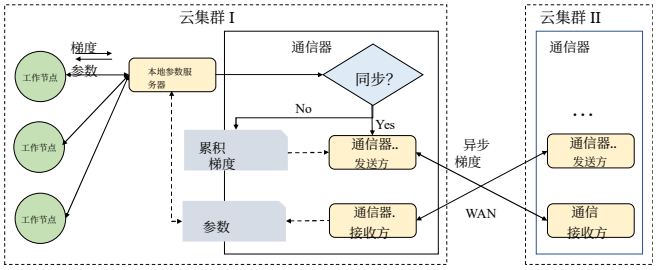

图 5. ASGD-GA同步过程

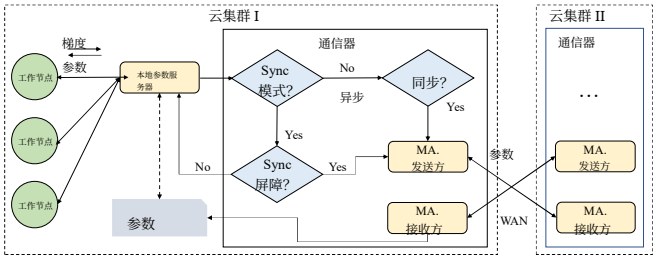

图 6. PS间MA同步过程

带梯度累积的异步随机梯度下降（ASGD-GA）。ASGD-GA 专注于同步梯度，如图 5 所示。其同步条件

定义为同步频率变量。待发送状态使用本地累积梯度。未达到同步条件时，梯度会通过合并新生成的梯度持续累积。通信模式采用异步方式，因此每个训练分区都无需阻塞等待其他分区准备好进行模型同步。接收方参数服务器采用随机梯度下降方法更新模型。

参数服务器间模型平均（MA）。在分布式机器学习系统中，模型平均是一种周期性同步方法，可以减少工作器之间的通信流量。在 Cloudless-Training 中，如图 6 所示，它用于减少不同云之间参数服务器的通信流量。MA 致力于改善模型参数处理体验。就发送模式而言，它可以是同步或异步的。对于待发送的模型状态，它使用本地模型参数。对于同步条件，在同步通信模式中使用同步屏障，在异步通信模式中使用同步频率变量。

II. 实现

Cloudless-Training 的实现采用 OpenFaaS[7] 作为基础无服务器运行时框架。开发可分为两部分，包括 OpenFaaS 针对多区域云计算的定制，以及工作流调整。

针对 OpenFaaS 的定制，我们进行了两项扩展。首先，增强 OpenFaaS，使其支持无服务器工作流部署和调用。在 OpenFaaS 中新增工作流这一实体，以便定义工作流的有向无环图。扩展 OpenFaaS 网关，使其能够识别工作流调用并调用内部工作流函数。其次，修改 OpenFaaS，使其支持函数寻址和识别。我们在 OpenFaaS 中维护一张函数寻址表，其中存储每个函数副本的身份、名称、命名空间和端点。这里的难点在于，函数的端点可能是动态的，因此映射也应实时更新。

对于工作流开发，其中包含控制平面和物理训练平面中的工作流。为构建控制平面，我们将调度器和通信器函数实现为有状态函数，并使用后端内存存储。为构建物理训练平面，我们迁移 ElasticDL[32]，使其适用于无服务器模式及相应的处理需求，例如通信。通信器函数通过 gRPC 发送方和接收方实现，以同步模型状态。对于底层训练，我们使用 TensorFlow。

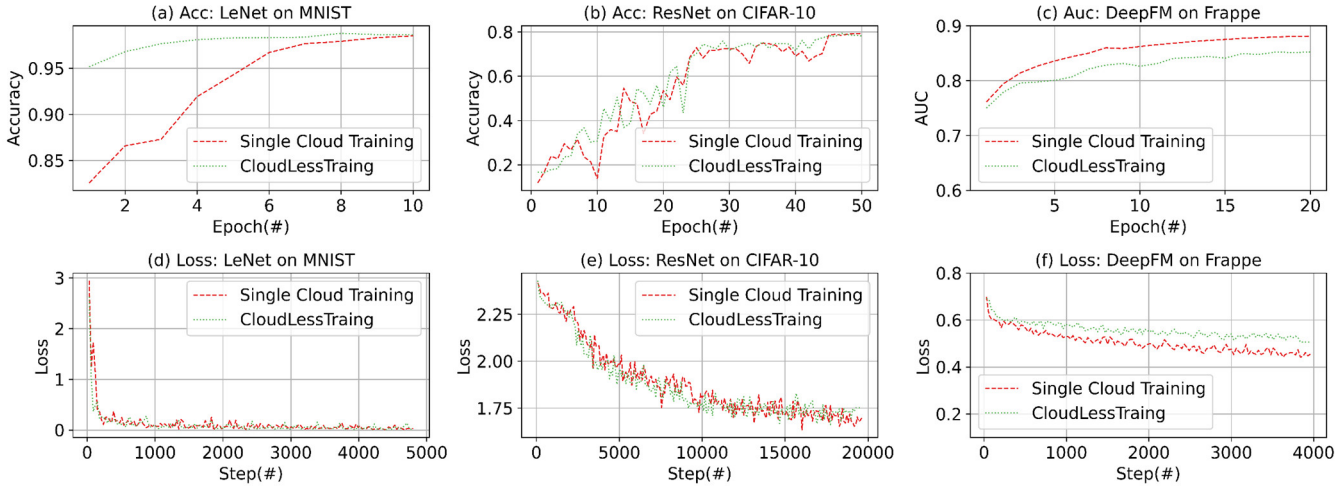

图 7. 在腾讯云上海区域和重庆区域使用 Cloudless-Training 训练 LeNet、ResNet 和 DeepFM 的准确率与损失对比，以及在腾讯云上海区域使用普通参数服务器进行机器学习训练的对比

III. 评估

我们在腾讯云这一中国典型公有云提供商上，使用 3 种人工智能模型对 Cloudless-Training 的易用性、弹性调度性能和通信性能进行了评估。结果表明，Cloudless-Training 能够以地理分布式方式支持通用机器学习训练，大幅提高资源利用率（例如，训练成本降低 9.2%–24.0%），并优化通信性能（例如，相较于基线，训练加速最高可达 1.7 倍），同时保证模型正确性。

环境。我们在腾讯云上海和重庆区域（分别位于中国东部和西部）构建了 2 个无服务器集群，并在其中部署了 Cloudless- Training。两个集群之间的广域网带宽为 100Mbps，这是腾讯云提供的最高配置。训练采用了 2 种中央处理器：英特尔至强 Cascade Lake（Cascade）和英特尔至强 Skylake（Sky）。

表 III. 实验模型和数据集

模型	梯度大小	数据集和大小	设置
LeNet	0.4MB	MNIST (60K)	轮次=10
ResNet	0.6MB	CIFAR-10 (50k)	轮次=50
DeepFM	2.4MB	Frappe（200k）	轮次=20

机器学习模型。实验使用 3 种人工智能模型进行评估，包括 LeNet[43], ResNet[6], 和 DeepFM[44], , 如表 III 所示。为降低在腾讯云上进行训练的货币成本，所用的 ResNet 模型是 ResNet18 的一个变体，其滤波器数量减少了 4 倍。基于这些模型，测量了特定轮次的训练时间等指标，

还测量了准确率、货币成本以及广域网上的通信时间。

A. 易用性

具体设置。为了评估 Cloudless- Training 的易用性，我们将其效果与在单个云中进行的简单机器学习训练进行比较。对于简单机器学习训练，在上海区域部署一个参数服务器框架，配备 24 个 CPU 核心（Cascade）和 48GB 随机存取存储器。对于 Cloudless-Training，每个区域配备 12 个 CPU 核心（Cascade）和 24GB 随机存取存储器。在这种情况下，两个框架拥有相同数量的云资源，并使用简单的异步随机梯度下降进行模型同步。通过测量准确率（或曲线下面积 - AUC）和损失来评估模型性能的正确性。

结果。如图 7 所示，在实验中，3 个模型均通过 Cloudless-Training 成功完成训练。在所有情况下，与常规机器学习训练相比，Cloudless-Training 的准确率（或曲线下面积）和损失收敛趋势相似。分别考虑 LeNet、ResNet 和 DeepFM 的训练准确率，Cloudless-Training 达到 0.9864、0.79、0.88、0.9851、0.78 和 0.84，接近常规机器学习训练的相应结果。就训练损失收敛速度而言，Cloudless- Training 在 LeNet 和 ResNet 上的表现与常规机器学习训练相当，而在 DeepFM 上略慢于常规机器学习训练。

B. 弹性调度的性能

表 IV. 弹性调度的资源规划设置

ID	***Data*** 分布 比率	基线计划 *(上海/重庆)*	资源规划 由算法 *(SH:CQ)*
1	1:1	级联/天空 12:12	级联/Sky 12:8
2	2:1	级联/级联 12:12	级联/级联 12:6
3	2:1	级联/Sky 12:12	级联/天空 12:4

具体设置。为评估弹性调度策略的性能，我们在上海（SH）和重庆（CQ）这两个区域采用了不同类型的中央处理器（分别为 Cascade 和 Sky），两者均可提供 12 个核心数大多数情况下。根据公式（1），两类资源的负载功率比约为 2:3。实验通过调整数据分布比例以及包含特定类型和核心数的资源规划，分为 3 组，如表 I 所示。所有基线实验都采用贪心策略，使用全部 24 个可用 CPU 核心，每个区域各 12 个。Cloudless-Training 的弹性调度策略生成的资源规划充分利用拖尾节点所在区域的 CPU 核心，并减少了另一区域的实际分配量。

结果。如图8和图9所示，通过弹性调度，地理分布式训练的资源利用率可显著提升，同时准确率趋势与基线相当，甚至更佳。

首先，如图8所示，资源利用率大幅提升，瞬时成本显著降低。图8(a-c)中的每一组都展示了采用弹性调度策略前后训练时间的占比（包括有效执行时间和等待拖尾节点的时间）。在每种情况下，总训练时间基本与基线相同，而等待时间有所减少。由于网络波动，总时间有时可能略高于基线。LeNet 的等待时间减少了 46.0%-82.6%，ResNet 减少了 82.3%-94.6%，DeepFM 减少了 6.8%-26.0%。由于 DeepFM 比另外两个模型更具通信密集型特征，其提升不如其他模型明显。随着资源利用率提高，等待掉队者期间因资源消耗产生的训练成本大幅降低。LeNet 的训练成本降低了 13.8%-16.0%，ResNet 降低了 9.2%-15.7%，DeepFM 降低了 13.4%-24.0%，如图8(d-f)所示。

其次，准确率和收敛速度均有所提升，如图9所示。大多数准确率曲线都略高于基线。此外，收敛速度大多快于基线，且波动较少。这是因为我们的调度有助于平衡各云之间的训练进度，从而减轻陈旧梯度的影响。同时，这也表明，采用我们策略训练的模型会更加稳定。

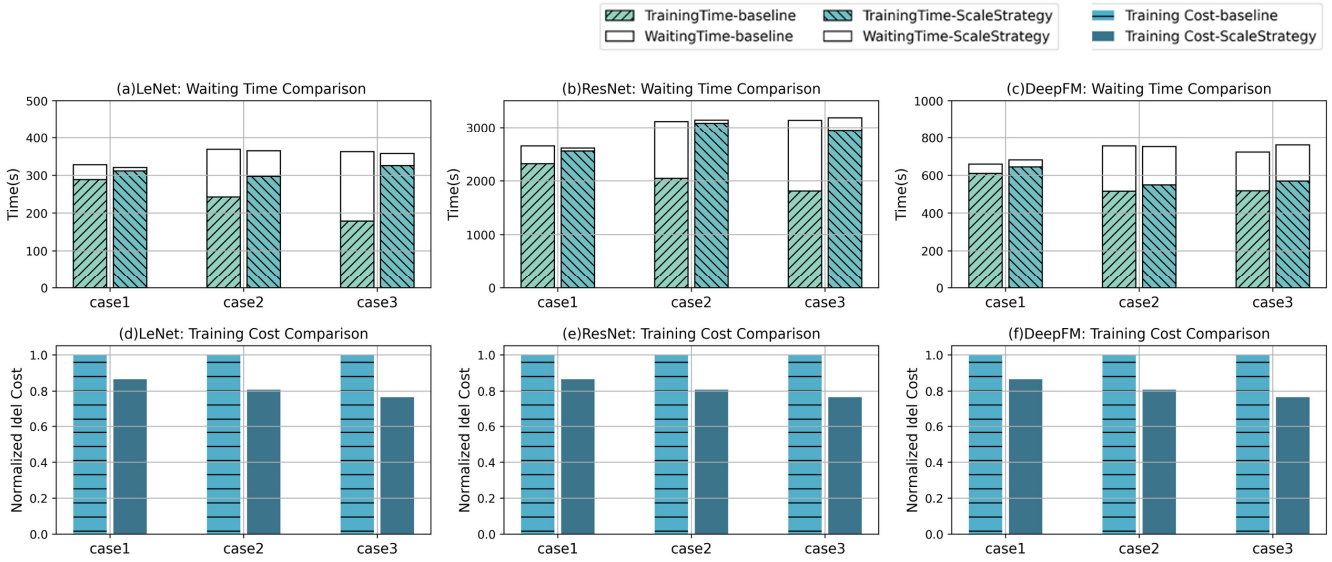

图8. 腾讯云上海和重庆区域中，3种不同数据分布（1:1、1:2、2:1）及 CPU 核心分配情况下，采用和不采用弹性调度的训练时间与成本对比

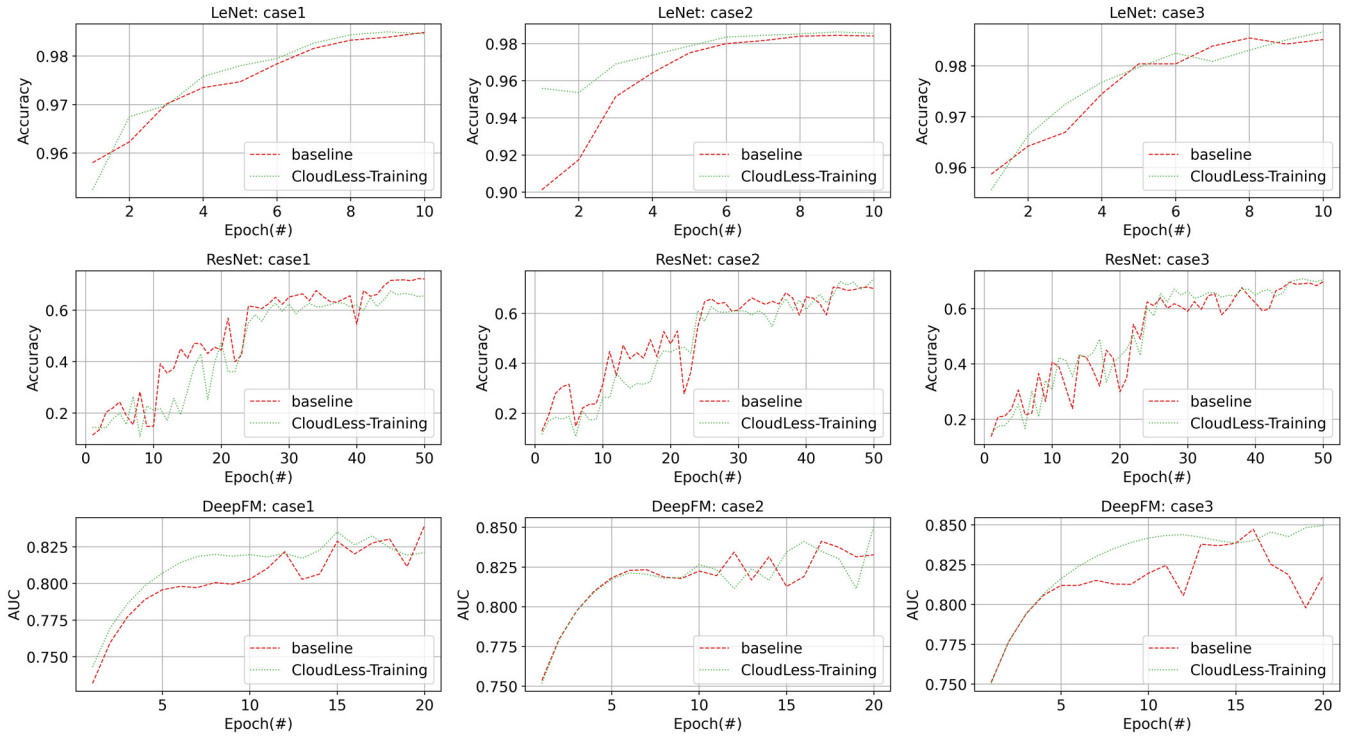

图9. 准确率收敛对比：在数据分布（1:1、1:2、2:1）和中央处理器分配各异的三种情况下，比较启用和未启用弹性调度的结果 res 腾讯云上海和重庆区域的分配情况

C. 同步性能

具体设置。为评估同步优化的效果，包括带梯度累积的异步随机梯度下降（ASGD-GA）和参数服务器间模型平均（MA），我们考察了4种模式：基线、ASGD-GA、采用异步模式的参数服务器间模型平均（AMA），以及采用同步模式的参数服务器间模型平均（SMA）。在基线模式下，各云中的训练通过简单的异步随机梯度下降相互同步。基线模式的同步频率为每1个迭代步一次。它可视为一种简单的多区域云版本的普通机器学习训练。对于ASGD-GA和AMA，我们分别测试了每4个和8个迭代步进行一次同步。除SMA外，其余模式均部署在前文提到的腾讯云上。SMA由于货币成本较高，因此在北京和上海的自托管云集群中进行测试。

结果。如图10和图11所示，通过Cloudless-Training 中的同步优化策略，可以显著提升地理分布式训练的性能。

首先，如图 10(a-c) 所示，ASGD-GA 和 AMA 都能有效加速地理分布式训练。LeNet、ResNet 和 DeepFM 的训练速度分别提升了 1.2×、1.2× 和 1.7× 。通信效率的提升主要源于云之间的同步频率从 1 次增加到 4 次、再到 8 次。这使总通信流量减少了数倍，而 ASGD-GA 和 AMA 的加速性能非常相似。当同步频率为 4 时，使用 ASGD-GA 训练 LeNet、ResNet 和 DeepFM 的通信时间比基线分别减少了 48.3%、52.7% 和 58.4%；使用 AMA 时分别减少了 46.0%、52.7% 和 58.5%。当同步频率增加到 8 时，使用 ASGD-GA 的时间比基线分别减少了 64.3%、70.1% 和 72.9%，使用 AMA 时分别减少了 60.8%、56.7% 和 71.0%。由于广域网（WAN）的波动，降幅并未达到理论预期的两倍。

其次，如图10(d-f)所示，Cloudless-Training 中的 ASGD-GA 和 AMA 均可确保地理分布式训练的精度收敛。在大多数情况下，ASGD-GA 和 AMA 的准确率变化趋势与基线相似。梯度延迟或模型同步延迟有时可能会减慢全局同步，但最终结果仍可达到甚至超过基线。

第三，SMA 的准确率最高，如图11所示。在自托管环境中比较基线、ASGD-GA、AMA 和 SMA，使用 SMA 的训练时间远慢于 ASGD-GA 和 AMA，与基线相近。然而，其准确率明显优于其他所有方法。我们认为，这种方法可用于提高训练准确率，值得进一步研究。

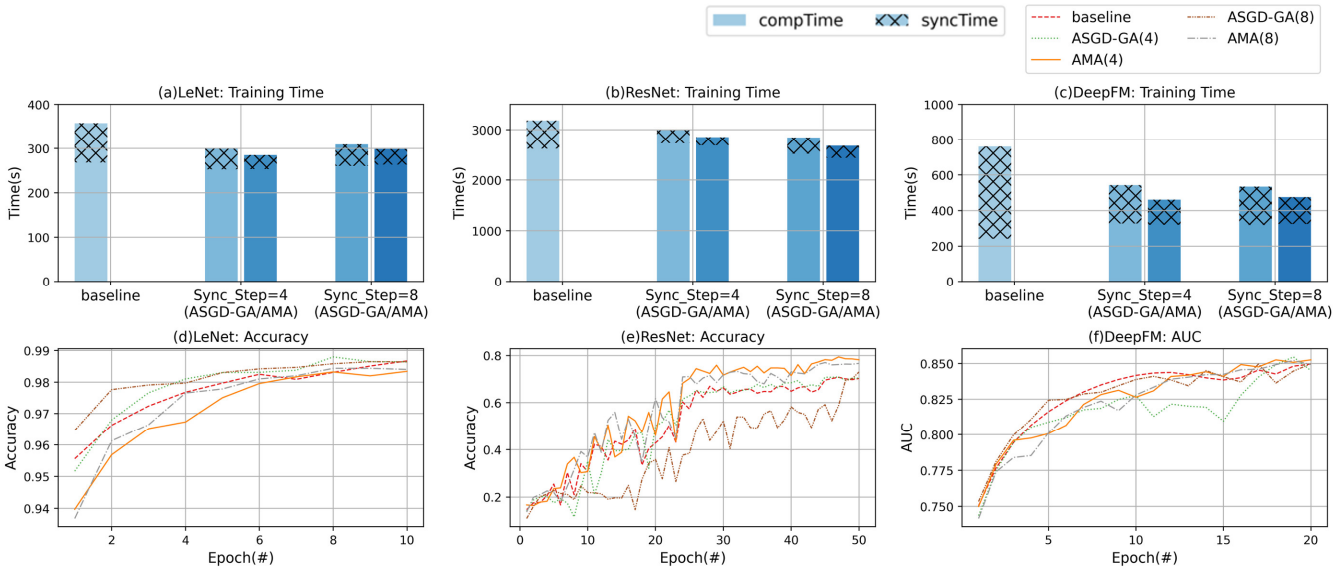

图10. ResNet的训练时间和精度收敛 在腾讯云上海和重庆区域，采用3种模型同步策略（ASGD、ASGD-GA、AMA）的ResNet准确率收敛对比

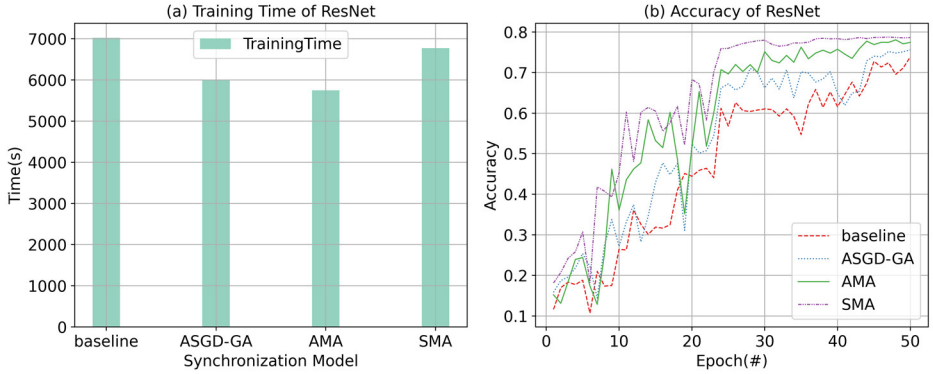

图11. 自托管环境中，采用4种模型同步策略（ASGD、ASGD-GA、AMA、SMA）的ResNet训练时间和准确率收敛对比

IV. 相关工作

地理分布式机器学习训练。近年来，地理分布式机器学习训练受到了广泛关注 [8][9][10]。它可以被视为分布式机器学习训练的一种高级变体，例如 Horovod[30] 和 Kungfu[31]。分布式机器学习训练中的一些挑战，在扩展到地理分布式机器学习训练时仍然存在，甚至更加突出。例如，通信问题在分布式和地理分布式机器学习训练中都受到广泛讨论。关于地理分布式机器学习训练的研究，主要是扩展基本分布式机器学习训练框架的机制（例如通信），并确保其基本可用性。Gaia[8] 使不同数据中心能够以接近局域网的速度进行通信。Nebula-I[9] 框架支持异构云集群协同训练深度学习模型。ElasticDL[32] 作为一种 Kubernetes 原生深度学习框架，天生适用于跨云训练，并支持容错和弹性调度。这些框架都没有像 Cloudless- Training 那样专注于解决负载均衡问题并探索各种通信优化策略。此外，Cloudless-Training 还采用了无服务器架构。

同步优化。它在分布式机器学习训练中得到了广泛研究，其中一些方法可以迁移到地理分布式机器学习训练中。工业界常用的同步策略有 3 种：（1）批量同步并行（BSP），要求工作器在每次计算后进行同步[47]。（2）陈旧同步并行（SSP），放宽了对同步频率的限制，允许较快的工作器比速度较慢的工作器多执行一定数量的迭代[48]。（3）无屏障异步并行（BAP），允许工作器在无需等待的情况下进行通信[49]。与此同时，学术界也提出了许多变体。近似同步并行（ASP）会持续发送梯度，直到达到显著性阈值[8]。去中心化并行随机梯度下降（D-PSGD）[46] 中，所有工作器都会在本地计算随机梯度，同时将各自的本地模型与邻居的模型取平均。异步去中心化并行随机梯度下降（AD-PSGD）[45] 则以异步模式将本地模型与邻居的模型取平均。

用于机器学习训练的无服务器计算。函数即服务 (FaaS) 已用于分布式机器学习训练，带来了更高的资源效率和扩展性能，但在地理分布式机器学习训练中尚未得到广泛采用。Kubeless[36], OpenWhisk[40], OpenFaaS[7] 和 Lambda[41] 等无服务器计算框架被用于实现事件驱动的弹性计算，并在许多特定应用场景中得到实践。对于简单的分布式机器学习训练，研究利用无服务器技术提升

并行训练弹性，例如 Cirrus[38], LambdaML[39]。LambdaML 提出了一种基于函数即服务的训练系统，用于确定函数即服务相较于基础设施即服务更具优势的情况。对于采用无服务器技术的地理分布式机器学习训练，例如 Fedless[37], 可实现跨大规模函数即服务提供商的联邦学习。在这项工作中，Cloudless-Training 定制了无服务器运行时，以支持地理分布式机器学习训练调度和通信，并构建了面向地理分布式机器学习训练的工作流。

V. 结论

本文介绍了 Cloudless-Training，这是一种采用无服务器架构的地理分布式机器学习训练框架。它采用两层架构，以无服务器的方式简化训练管理的逻辑视图，涵盖调度和模型同步。在调度方面，该框架提供了一种弹性调度策略，通过对可用云资源的异构性以及现有训练数据集的分布进行建模，确保均衡的负载和资源配置。在模型同步方面， 它基于 WAN 通信寻址支持，构建了云之间的基本广域网通信机制，并提供了两种策略：带梯度累积的异步随机梯度下降（ASGD-GA）和参数服务器间模型平均（MA）。实验表明，Cloudless-Training 能够支持以地理分布式方式进行通用机器学习训练，大幅提高资源利用率（例如，训练成本降低 9.2%–24.0%）和通信性能（例如，大多数训练任务相较于基线最多提速 1.7 倍），同时保证模型正确性。因此，我们认为 Cloudless-Training 是一种实用有效的系统，能够提升地理分布式机器学习训练的效率。

参考文献

[1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal 等。“语言模型是少样本学习者。”《神经信息处理系统进展》，33：1877-1901，2020年。[2] L. Li, Y. Fan, M. Tse 和 K.Y. Lin。“联邦学习应用综述”。《计算机与工业工程》，149(5)，106854，2020年。[3] J. Zhang, Z. Qu, C. Chen, H. Wang, Y. Zhan, B. Ye 等。“边缘学习：边缘分布式大数据分析的使能技术。”《ACM计算调查》（CSUR）第54卷，36，2021年，7，1，第 - 页。[4] M. Zhang, C. Krintz 和 R. Wolski。“Stoic：面向边缘设备机器学习应用的无服务器可远程操控混合云。”2020年IEEE普适计算与通信国际会议研讨会（PerCom Workshops）。IEEE，2020年。[5] （2022年）腾讯云。[在线。] 网址：https://cloud.tencent.com/?mobile&lang=en。[6] K. He, X. Zhang, S. Ren 和 J. Sun。（2016年）。用于图像识别的深度残差学习。IEEE。[7] （2019年）。OpenFaaS——让无服务器函数变得简单。https://github.com/openfaas/faas。[8] K. Hsieh, A. Harlap, N. Vijaykumar, D. Konomis, G. R. Ganger 和 P. B. Gibbons。“Gaia：以局域网速度实现地理分布式机器学习。”第14届USENIX网络系统设计与实现研讨会（NSDI 17）。2017年。[9] Y. Xiang, Z. Wu, W. Gong, S. Ding, X. Mo, Y. Liu。“Nebula-I：在低带宽云集群上协同训练深度学习模型的通用框架。”arXiv预印本 arXiv:2205.09470，2022年。[10] C. C. Hung, G. Ananthanarayanan, L. Golubchik, M. Yu 和 M. Y. Zhang。“利用多种资源进行广域分析。”第13届EuroSys会议论文集。2018年。[11] Shi, S., X. Zhou, S. Song, X. Y. Wang, Z.L. Hu, X. Huang 等。“面向公有云集群上深度学习的可扩展分布式训练。”《机器学习与系统会议论文集》3（2021年）： 401-412。

[12] M. Li、L. Zhou、Z. Yang、A. Li、F. Xia、D.G. Andersen 和 A. J. Smola。《分布式机器学习的参数服务器》。NIPS 大学习研讨会。第 6 卷，第 2 期。2013年。[13] Y. Lin、S. Han、H. Mao、Y. Wang 和 W. J. Dally。《深度梯度压缩：降低分布式训练的通信带宽》。arXiv 预印本 arXiv:1712.01887（2017年）[14] H. Zhang、Z. Zheng、S. Z. Xu、W. Dai、Q. R. Ho、X. D. Liang 等。《Poseidon：面向 GPU 集群上分布式深度学习的高效通信架构》。2017年 USENIX 年度技术会议（USENIX ATC 17）。2017年。[15] N. Shazeer、Y. L. Cheng、N. Parmar、D. Tran、A. Vaswani、P. Koanantakool 等。《Mesh-tensorflow：面向超级计算机的深度学习》。神经信息处理系统进展第 31 卷（2018年）。[16] D. Narayanan、A. Phanishayee、K. Shi、X. Chen 和 M. Zaharia。《内存高效的流水线并行 DNN 训练》。国际机器学习会议。PMLR，2021年。[17] M. I. Jordan 和 T. M. Mitchell。《机器学习：趋势、观点与前景》。《科学》349.6245：255-260，2015年。[18] V. Sze、Y. J. Chen、H. T. Yang 和 J. S. Emer。《深度神经网络的高效处理：教程与综述》。IEEE 论文集 105.12：2295-2329，2017年。[19] B.、Hind、S. Rakrak、S. Raghay 和 B. Buhnova。《迈向边缘云物联网：最新进展与未来研究方向》。《电子学》7(11):309，2018年。[20] N. Khan、I. Yaqoob、I. A. T. Hashem、Z. Inayat、W. K. M. Ali、M. Alam 等。《大数据：综述、技术、机遇与挑战》。《科学世界期刊》，712826-712826，2014年。[21] S. Mishral 和 A. K. Tyagi。《机器学习技术在基于物联网的云应用中的作用》。《基于人工智能的物联网系统》。施普林格，Cham，105-135，2022年。[22] （2022年）TikTok 服务器位置引发质疑，美国联邦通信委员会要求应用商店下架该应用。[在线。] 网址：https://marketrealist.com/p/where-are- tiktok-servers-located。[23] （2022年）Amazon EC2 实例网络带宽 [在线。] 网址：https://docs.aws.amazon.com/zh_cn/AWSEC2/latest/UserGuide/ec2- instance-network-bandwidth.html。[24] G. McGrath 和 P. R. Brenner。《无服务器计算：设计、实现与性能》。2017年 IEEE 第37届分布式计算系统国际会议研讨会（ICDCSW）。IEEE，2017年。[25] H. Cui、H. Zhang、G. R. Ganger、P. B. Gibbons 和 E. P. Xing。《Geeps：通过 GPU 专用参数服务器实现分布式 GPU 上的可扩展深度学习》。第11届欧洲计算机系统会议论文集。2016年。[26] S. Ruder。《梯度下降优化算法概述》。arXiv 预印本 arXiv:1609.04747，2016年。[27] Y. Hao、S. Yang 和 S. Zhu。《并行重启 SGD：更快收敛、更少通信——揭示模型平均为何适用于深度学习》。AAAI 人工智能会议论文集。第 33 卷，第 01 期。2019年。[28] K. Alex 和 G. Hinton。《从微型图像中学习多层特征》（技术报告）。多伦多大学，2009年。[29] M. Shoeybi、M. Patwary、R. Puri、P. LeGresley、J. Casper 和 B. Catanzaro。《Megatron-LM：使用模型并行训练数十亿参数的语言模型》。arXiv 预印本 arXiv:1909.08053（2019年）。[30] S. Alexander 和 M. D. Balso。《Horovod：在 TensorFlow 中快速、轻松地进行分布式深度学习》。arXiv 预印本 arXiv:1802.05799，2018年。[31] L. Mai、G. Li、M. Wagenländer、K. Fertakis、A. Brabete 和 P. Pietzuch。《KungFu：使分布式机器学习训练具备自适应能力》。第14届 USENIX 操作系统设计与实现研讨会（OSDI 20）。2020年。[32] （2019年）Kubernetes 原生深度学习框架 [在线。] 网址：https://github.com/sql-machine-learning/elasticdl。[33] M. Luo、C. Hong 和 P. Costa。《优化分布式机器学习中的网络性能》。第7届 USENIX 云计算热点专题研讨会（HotCloud 15）。2015年。[34] （2017年）Baidu-allreduce [在线。] 网址：https://github.com/baidu- research/baidu-allreduce。[35] A. A. Fikri 和 K. Heafield。《分布式梯度下降中的稀疏通信》。arXiv 预印本 arXiv:1704.05021，2017年。

[36] （2019年）Kubeless：Kubernetes 原生无服务器框架。[在线。] 网址：https: //kubeless.io。[37] A. Grafberger、M. Chadha、A. Jindal、J. Gu和 M. Gerndt。《FedLess：利用无服务器计算实现安全、可扩展的联邦学习》。2021年 IEEE大数据国际会议（Big Data）。IEEE，2021年。[38] J. Carreira、P. Fonseca、A. Tumanov、A. Zhang 和 R. K.。《Cirrus：端到端机器学习工作流的无服务器框架》。2019年 ACM云计算研讨会（SoCC ’19），2019年。[39] J. Jiang、S. Gan、Y. Liu、F. Wang、G. Alonse、A. Klimovic 等。《迈向揭示无服务器机器学习训练的奥秘》。2021年 ACM数据管理国际会议（SIGMOD ’21），2021年。[40] （2017年）Apache OpenWhisk。[在线。] 网址：http://openwhisk.org/。[41] （2022年）Lambda。[在线。] 网址：https://aws.amazon.com/lambda/。[42] S. Rashidi、M. Denton、S. Sridharan、S. Srinivasan、A. Suresh、Jade Nie 等。《实现分布式深度学习训练平台中的计算与通信重叠》。2021年 ACM/IEEE 第48届国际计算机体系结构研讨会（ISCA）。IEEE，2021年。[43] Y. Le Cun、B. Boser、J. S. Denker、D. Henderson、R. E. Howard、W. Hubbard。《使用反向传播网络进行手写数字识别》。神经信息处理系统进展 2，1989年。[44] H. Guo、R. Tang、Y. Ye、Z. Li 和 X. He。《DeepFM：一种基于分解机的点击率预测神经网络》。arXiv 预印本 arXiv:1703.04247，201年。[45] X. Lian、W. Zhang、C. Zhang 和 J. Liu。《异步去中心化并行随机梯度下降》。国际机器学习会议。PMLR，2018年。[46] X. Lian、C. Zhang、H. Zhang、C.-J. Hsieh、W. Zhang 和 J. Liu。《去中心化算法能否胜过集中式算法？去中心化并行随机梯度下降的案例研究》。神经信息处理系统进展 30。2017年。[47] E. P. Xing、Q. Ho、P. Xie 和 D. Wei。《大数据上的分布式机器学习策略与原则》。Engineering 2，2（2016年），179-195。2016年。[48] J. Verbraeken、M. Wolting、J. Katzy、J. Kloppenburg、T. Verbelen 和 J. S. Rellermeyer。《分布式机器学习综述》。ACM计算调查（CSUR）53.2（2020年）：1-33。2020年。[49] H. Minyang 和 K. Daudjee。《Giraph Unchained：类Pregel图处理系统中的无屏障异步并行执行》。VLDB学报 8.9（2015年）：950-961。2015年。