中兴通讯技术
ZTE Technology Journal
ISSN 1009-6868,CN 34-1228/TN

《中兴通讯技术》网络首发论文

题目：　新型拓扑感知的参数交换方案
作者：　万鑫晨，胡水海，张骏雪
网络首发日期：　2020-10-10
引用格式：　万鑫晨，胡水海，张骏雪．新型拓扑感知的参数交换方案．中兴通讯技术．
https://kns.cnki.net/kcms/detail/34.1228.TN.20201009.1536.002.html

新型拓扑感知的参数交换方案

New Parameter Exchange Scheme with Topology-awareness

万鑫晨/WAN Xinchen

胡水海/HU Shuihai

张骏雪/ZHANG Junxue

（香港科技大学，中国 香港 999077）

(Hong Kong University of Science and Technology, Hong Kong SAR 999077, China)

摘要：

定义了一种新型拓扑感知的参数交换方案——RAT。针对底层物理拓扑及其超额认购条件，RAT 建立了“弹性全局规约树”。该树指定了参数聚合模式，其中每个聚合节点负责在规约阶段聚合一个超额认购区域内的所有工作的梯度，并在广播阶段将更新传回给工作节点。实验表明，该方法能有效地减少跨超额认购区域流量，缩短依赖链。

关键词：

分布式机器学习；全局规约算法；参数交换方案

Abstract:

A new solution called RAT that determines the communication pattern for distributed machine learning (DML) is proposed. Aiming at the underlying physical topology and its oversubscription conditions, RAT establishes the "resilient allreduce trees". The allreduce trees specify the aggregation pattern in which each aggregator is responsible for aggregating gradients from all workers within an oversubscribed region at the reduce phase, and broadcasting the updates back to workers at the broadcast phase. Experiments show that this method can effectively reduce the cross-region traffic and shorten dependency chain.

Keywords:distributed machine learning; all-reduce algorithm; parameter exchange scheme

近年来，深度神经网络（DNN）被广泛应用于计算机视觉、自然语言处理等多个应用领域。由于 DNN 训练任务可能需要数天或数周才能完成，为了缩短训练时间，分布式机器学习系统被引入 DNN 训练过程。因此，大量关于分布式机器学习（DML）系统加速训练的研究和方法在学术界和工业界不断涌现。

由于 DML 是计算密集型任务，之前大部分的研究主要集中在为集群计算资源设计高效的调度策略上。然而，随着图形处理器（GPU）算力的逐步提升和模型尺寸的增大，我们发现整体的训练性能瓶颈逐渐从计算部分转移至通信部分。例如，当在 32 GPU 集群中（如 VGG16 的大模型）训练时，通信部分的完成时间占据训练任务总完成时间的 90%[1]。当前已经出现了大量利用 DML 训练的鲁棒性，在参数同步机制[2]和减少网络通信量[3]等方面来减缓 DML 通信瓶颈的研究成果，以及利用传统数据中心网络的流调度[4-7]和协同流调度[8-10]技术来进行通信优化的研究成果。本文中，我们主要研究 DML 中的参数交换过程。参数交换过程由预先设置好的参数交换方案来定义，该方案描述了每轮迭代中的参数/梯度交换方式。考虑到 DNN 通常需要经过成百上千次的迭代训练，因此针对参数交换方案的研究和优化可能会带来潜在巨大的性能提升。常见的参数交换方案有参数服务器（PS）和环形全局规约（Ring）等，这些参数交换方案现均已在各主流通用深度学习框架下成功实现并部署。专业人士评测后表示，这些方案在常规网络场景中为分布式机器学习任务提供了良好的参数交换性能。然而，我们认为，PS 和 Ring 等方案在某些存在故障或不确定性事件的网络场景下（例如超额认购网络和存在故障的网络）存在严重的性能下降问题。事实上，在大规模数据中心网络内部，存在诸多类似事件发生的可能情况，例如节点故障、突发流量淹没交换机或网卡、网络 incast 现象等。当前常见方案均无法适应这类网络场景，因此，设计并实现新型参数交换方案以适应这类存在故障和不确定性事件的数据中心网络场景，具有重大的研究和应用价值。

1 背景介绍

1.1 数据中心网络

数据中心网络（DCN）通常采用多层树状拓扑结构。如图 1 所示，在这种拓扑中，交换机按层划分并树状连接（通常是 2 层或 3 层结构）。服务器在拓扑叶端与机架顶部交换机（ToR）直接相连并对应分组。多层树状拓扑结构为 DCN 的搭建和扩展带来极大的便利性和灵活性，系统架构人员可以通过在每层简单地增加交换机数量和交换机与服务器之间的网络连接，来扩展网络规模。

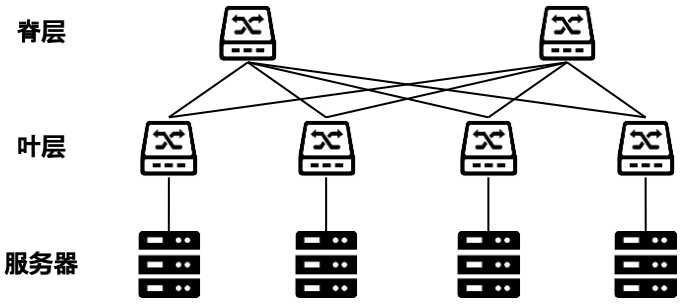


图 1 两层脊-叶结构数据中心网络拓扑

然而，DCN 存在若干故障和不确定性事件，包括超额认购事件、网络拥塞和故障问题。为了降低搭建 DCN 所需的昂贵成本[11]，研究人员引入了超额认购的概念，即利用各源端服务器很少同时进行大规模数据传输的特性，使得终端服务器流入 DCN 的最大理论流量略大于网络最大可承载量（通常超额认购比率在 4∶1 和 8∶1 之间[12]）。通过这种方式可以有效减少交换机和网络连接数量，从而降低 DCN 搭建成本。然而，超额认购是一把双刃剑：一方面，它在不增加 DCN 搭建成本的前提下有效地增大了集群规模；另一方面，它在某些情况下，如多主机并发传输大规模流量等，会给 DCN 带来巨大网络拥塞风险。当网络实时总通信量超过某特定阈值时，网络中枢部分（即核心交换机等）就会发生网络拥塞。最坏的情况是会损坏网络中枢部分的数据传输能力，造成整个网络无法提供数据传输的后果。此外，网络拥塞现象可能会在出现突发流量淹没某些链路或网卡时，或当低优先级流量在交换机上被持续到来的高优先级抢占传输等情况下发生。网络故障现象可能发生在物理层，例如物理链路故障、节点故障等。

1.2 分布式机器学习

数据并行是分布式机器学习最常用的并行模式。如图 2 所示，每个工作节点负责维护自己的本地模型，并独立地基于与其他节点互不重叠的一部分数据集进行训练。训练过程以迭代的方式完成，其中每轮迭代包含两个阶段：第一阶段是计算密集型的本地模型训练阶段，包括前向传播生成对小批输入的预测，反向传播导出与预测和目标标签之间的损失相关的局部梯度；第二阶段是通信密集型的参数交换阶段，在该阶段通过对所有局部梯度取平均值的方式来计算平均梯度，并将结果输入到优化器中以更新全局模型参数。更新后的参数被发回给每个工作节点，然后工作节点使用更新后的模型版本以开始下一轮迭代过程。

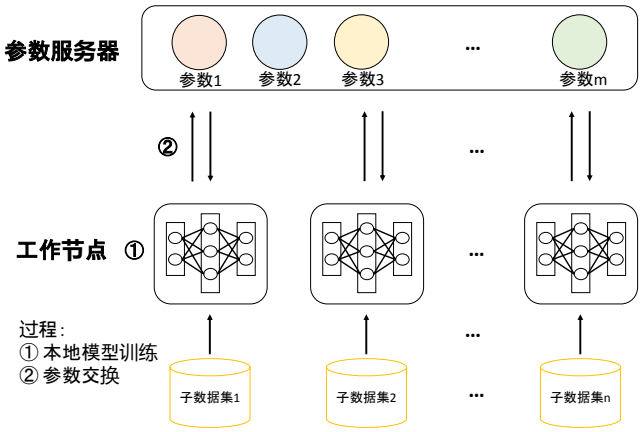


图 2 分布式机器学习工作流程

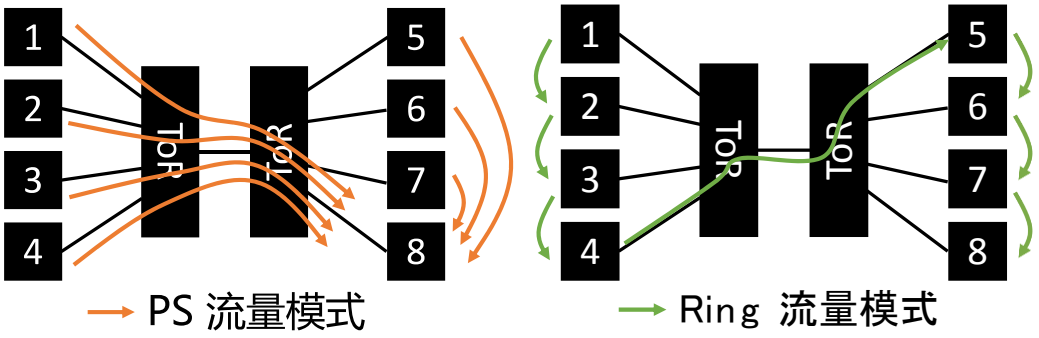
上述参数交换阶段通常遵循批量同步并行（BSP）的同步模式，这是因为它能提供最佳的机器学习模型预测性能，保证任务的可再现特性。因此，BSP 成为当前最主流的同步模式。在该模式下，所有的工作节点在每轮迭代中都需要完成全局同步，随后才能开始新一轮的迭代。

2 现有的参数交换方案

每个任务的参数交换阶段均执行着一套特定的参数交换方案，该方案描述了在每轮迭代中服务器之间的逻辑参数交换过程。在这里，我们对 DML 任务的一些常用参数交换方案进行分类，并讨论它们各自的局限性。

2.1 PS 方案

该方案已被应用于 TensorFlow[13]、Caffe[14]、MXNet[15]等多个流行 DNN 框架中。PS 采用了一种直接通信模式，其中参数在工作节点和 PS 间直接同步。工作节点在计算并生成局部梯度后，将其直接推至 PS，并在 PS 完成聚合过程后将更新的模型参数拉取回来。



（a） PS 流量模式　　（b）Ring 流量模式

PS：参数服务器

Ring：环形全局规约

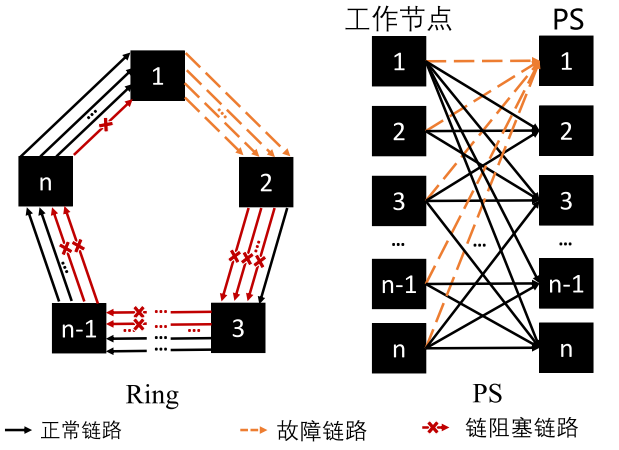
ToR：机顶交换机

图 3 PS 和 Ring 的流量模式

尽管 PS 方案直接有效，但并不适用于存在超额认购的网络环境。图 3（ a）展示了一个 PS 流量模式的示意图。假设工作节点和 PS 同时被放置在每个节点上，我们观察到，跨机架的链路相较于每条机架内的链路将承受额外近 1.3 倍的流量负载。对于给定集群配置（包括机架 r、w 工作节点和集群超额认购比率 o），平均任务完成时间将会有$\frac{o(w-1)}{w\left(1-\frac{1}{r}\right)}$倍性能下降。这意味着对于节点数量较多的大型作业，跨机架链路与机架内链路的流量不均衡问题会变得更加严重，我们在第 5 章中的仿真实验也验证了这一推论。需要注意的是，服务器在每个机架上的摆放位置并不会缓解这一问题。这是因为对于一个给定规模的集群，机架间的通信不会改变；而关键因素是 PS 采用的直接通信模式。

2.2 Ring 方案

Ring 方案已应用于 BaiduRing[16]和 Horovod[17]等。DNN 训练开始时，每个节点两两顺次相连组成环状拓扑；在之后的参数交换阶段，各节点保持同一圆周方向传输梯度。Ring 方案对应的参数交换过程可分为两个阶段：scatter-reduce 和 all-gather。以逆时针方向进行 scatter-reduce 为例，生成本地梯度更新后，每个工作节点从它的左手边接收一个梯度块，与它的本地梯度块进行聚合，并将聚合结果块发送给右边的工作节点。重复上述过程 $n-1$ 轮后，每个工作节点中各有一个聚合了所有工作节点本地梯度的梯度块。在 all-gather 阶段中，n 个工作节点简单在每轮迭代中复制接收到的对应位置梯度块，并重复 $n-1$ 次上述操作，从而完成整个参数交换阶段。



总链路利用率	1	2	…	N	平均 (n趋近无穷大)
Ring	0	1/n	…	1-1/n	50%
PS	0	(n-1)/n	…	(n-1)/n	100%

n：节点数

PS：参数服务器

Ring：环形全局规约

图 4 Ring 存在“链阻塞”现象

与 PS 相比，Ring-allreduce 在每一跳均进行梯度聚合，因此实现了最小化跨机架流量负载（见表 1）。与此同时，它引入了太多的节点间依赖关系，很容易造成网络拥塞或故障。如图 4 所示，*n* 个节点参与以进行环形全局规约。我们假设，某时刻节点 1 暂时不能向 2 发送数据，那么造成这种现象的原因可能有很多种：例如 1 和 2 之间的链路出现故障，或是该条链路发生拥塞，或是链路带宽优先分给了其他流量，或者该节点本身出现故障等。在这种情况下，节点 2 只能通过其中一条链向节点 3 发送 1/*n* 的数据，因为 *n*−1 条链在节点 1 处被阻塞了。接着，节点 3 只能向节点 4 发送 2/*n* 的数据，依此类推。这种节点依赖性会对所有下游节点产生级联效应。当 *n* 较大时，会导致整体 50%的网络利用率下降。我们将这种现象称为“链阻塞”，在第 5 章中我们的仿真结果也将揭示它的影响。与之相反，PS 不会遇到这个问题，因为其所使用的直接通信模式仅引入最小的依赖性。

2.3 其他集合全局规约方案

其他集合全局规约方案，如 k-nominal tree[18]、butterfly mixing[19]和 recursive halving and doubling[20]，均可以视为综合了 PS 和 Ring 的方案，这些方案具有预先确定的参数交换模式，然而这些模式对底层网络拓扑不可知；因此，它们在某种程度上同样存在与 PS 和 Ring 类似的问题，例如，跨机架的额外通信流量和长链节点依赖关系。我们在表 1 中分别列出了它们各自对应的值，并强调了其局限性。

此外，最近的一些全局规约方案[21-24]是通过感知分层网络拓扑来执行梯度聚合。然而在大规

模网络环境下，它们或多或少面临着一些问题。BlueConnect[21]依照网络拓扑的区域划分，将集群的大环分解为对应多个区域的小环。相较于传统的 Ring，它以一种更细粒度的方式运行，并减轻了由环中最慢的链路带来的影响。由于它是一种基于 Ring 的变体方案，因而也继承了 Ring 的脆弱性。当每个机架规模增大时，BlueConnect 的运行情况会变差。HiPS[22]采用 RDMA（远程直接数据存取技术）传输来进行全局规约，它特别适用于以服务器为中心的一类网络拓扑；然而当它在 Ring 模式下运行时，会引入额外的依赖链。ParameterHub[23]是一种协同设计软硬件的参数交换方案，其核心是 PBoxes（一台配备了 10 块网卡的服务器）在机顶交换机（TOR）中被用来减少跨机架的通信量。然而，它引入了额外的硬件特殊偏好（每台服务器上配置多块用于聚合的网卡），并且不能保证最小跨区域通信量。Plink[24]依据网络拓扑应用了一个 2 级的层次结构聚合，然而当网络层次结构超过 2 时，会产生同样的额外跨机架流量问题。

3 新型拓扑感知的参数交换方案设计

第 2 章中讨论的各方案的局限性启发了我们定义参数交换方案的期望属性：

- 实现最小跨超额认购区域（如机架、pod）流量，以避免造成网络瓶颈；
- 短依赖关系链，以更好地弹性应对网络拥塞和故障；
- 结构简单，以减小因引入参数交换方案而带来的必要计算和执行开销。

云星科技基于上述期望属性的定义，设计并实现了一套具备拓扑感知能力的参数交换方案——RAT（弹性全局规约树）。

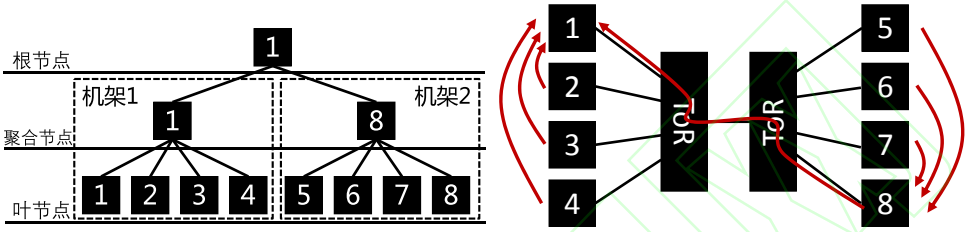
3.1 RAT 的主要角色

对于一个给定的物理网络拓扑 *T*，我们为 DML 任务 *J* 以一种简单的分层结构构建 RAT 树，构建的同时会考虑到超额认购区域（如机架、Pod）。树上每个节点扮演以下一个或多个角色：

- 叶节点：负责发送它的局部梯度或接收全局的更新参数。任务 *J* 中的每个工作节点均对应 RAT 上的一个叶节点。
- 聚合节点：对于拓扑 T 中的每个超额认购区域，RAT 引入了相应的聚合层，从而使跨区域流量最小化。在规约阶段，每个聚合节点负责将区域内的所有叶节点或下层聚合节点上的梯度更新以进行聚合，并将聚合后的梯度更新发送到上层聚合节点或根节点。在广播阶段，上述过程以逆方向运行。
- 根节点：负责聚合全局所有梯度，计算全局更新，并以相反的方向返回给下层聚合节点或叶节点。

3.2 RAT 的构建算法

RAT 将全部节点划分为不同组，并依据拓扑结构进行分层，按层聚集梯度。整体聚合过程如下：在规约阶段，对于最底部的叶子层，RAT 算法为每个物理机架，即超额认购区域，各分配一个 0 级聚合节点，该节点负责聚合同一机架内的所有梯度更新；然后在上一层的每个超额认购区域中，从区域内的所有 0 级聚合节点中指定一个 1 级聚合节点，来负责聚合区域内所有 0 级聚合结点的梯度更新；之后对更上层的拓扑节点聚合，同样遵循相同的例程，直到所有最初来自叶子的梯度都聚合到一个（n-1）级的聚合节点中，该节点也被称为根节点。之后，广播阶段开始并且将以上操作反向分层进行。



(a) RAT 树层次图　　(b) RAT 流量模式

TOR：机顶交换机

图 5 图 3 拓扑下的 RAT 树及其流量模式

图 5 展示了一棵基于 2 机架 8 节点的网络拓扑建立的 RAT 树，其中每个工作节点对应一个叶节点，并且某些工作节点被指定了不同级的聚合节点或根节点角色，从而使这棵 RAT 树能够在指定的网络拓扑中执行高效的参数聚合任务。需要提醒的是，在该例中我们只是简单考虑了机架级的超额认购场景和机架级的聚合节点。实际上，RAT 适用于所有树形数据中心网络拓扑场景。此外，尽管在这个拓扑中，我们按照根节点指定的不同，总共可以组成 8 个不同 RAT。假设网络具有对称性，且每个 RAT 上承载完全相同大小的工作负载，我们将网络流量均匀地分布在每个 RAT 上，以实现集群的负载均衡。我们将在非对称网络拓扑场景下每棵 RAT 树的流量负载非均匀分配问题留为以后的研究工作。

3.3 RAT 的属性

表 1 显示了 RAT 与其他主流参数交换方案在最小化跨区域流量和节点依赖链长这两个参数上的对比，从而说明了 RAT 完全满足上述的参数交换方案的期望属性。表 1 中的几个参数分别为：l 表示超额认购层数，w 表示总工作节点数量，w_r表示每个机架中的工作节点数量。

RAT 满足全部期望属性的原因主要有：首先，除了 Ring 和 BlueConnect 之外，其他所有的主流方案都不能最小化跨超额认购区域的流量。相反，由于 RAT 是为网络拓扑专门定制的，

因此它通过为每个超额认购区域引入一个聚合节点，来优化整体跨区域流量大小。其次，RAT 引入了一个长度为 2（|l| + 1）的节点依赖链。由于数据中心网络通常超额认购层数较少（例如 1 或 2），该链长通常小于除 PS 以外的其他所有方案。另外，RAT 采用一种简单且规律的结构和一套容易实施的构建树算法，简单规律的结构仅包含 3 种不同角色，构建树算法依据网络拓扑递归构建树，从而极大地简化了计算和执行参数交换过程。

表 1 各参数交换方案关于跨区域流量及依赖链长的对比

参数比较	PS	Ring	Butterfly	Halving& Doubling	K-nominal	BlueConnect	PLink	RAT
最小跨区域流量	×	✓	×	×	×	✓	×	✓
依赖链长	2	2(|w|-1)	$\log_2(\|w\|)$	$2\log_2(\|w\|)$	$2\log_k(\|w\|)$	2(|l|+max$\{w_r\}$)	4	2(|l|+1)

w_r：每个机架中的工作节点数量

Butterfly、Halving& Doubling、K-nominal、BlueConnect、Plink：均为方案名

l：超额认购层数

PS：参数服务器

RAT：新型拓扑感知的参数交换方案

Ring：环形全局规约

w：总工作节点数量

4 相关实验

在本节中，我们将 RAT 分别与两种有代表性的参数交换方案——PS 和 Ring 进行仿真实验对比，来量化展示 RAT 在网络拥塞及故障等场景下具备的弹性适应能力。

4.1 仿真实验设置

我们在仿真中使用了两种不同的实验设置。在超额认购场景中，我们使用配备了 2 个 spine

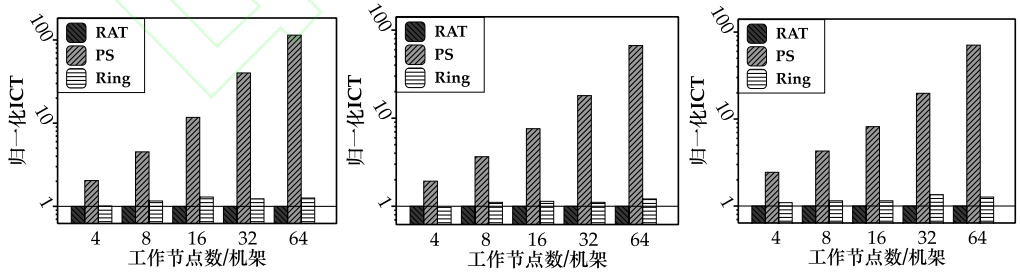
交换机和 4 个 leaf 交换机的传统 spine-leaf 网络拓扑，并将每个机架上的工作节点数量设为变量，从而使网络超额认购比率也随之变化（从 2∶1 变为到 32∶1）。在网络存在故障的场景中，我们在网络容量充足（即无超额认购）的 2 机架 64 台服务器且链路带宽均为 40 Gbit/s 的集群上运行机器学习任务流量。我们通过暂停一些节点发送数据来模拟网络中某些节点或链路发送拥塞或故障的现象，即从某时刻开始随机选择 k 个节点以暂停发送数据，并在每隔 d 时间周期性地随机改变这 k 个节点。此外，我们通过测量每个任务的单轮迭代完成时间（ICT）来评价该任务的训练性能。

4.2 流量模式设置

我们在 NS3（网络模拟器）中模拟了 PS、Ring 和 RAT 的参数交换模式。对于 PS，我们将 PS 和工作节点设置为同在每台服务器中，并通过以多对多发送相同大小数据的形式模拟 PS 下的参数交换过程。对于 RAT，按照 RAT 的算法构建了 n 棵 RAT 树，其中 n 为集群中总节点数量，且集群中的每个节点恰好对应每棵 RAT 树的根节点。我们将总通信量均匀地分布在每棵 RAT 树上以实现负载平衡。对于 Ring 的模拟，将集群中的所有节点两两连接成逻辑环，并仅允许每个节点与其邻居进行通信。将单轮迭代的总网络通信量大小设置为与 ResNet50 相同（总计 97 MB），并在 3 种模式下分别启动流量发生器。需要提到的是，为了简单起见，假设计算和通信之间是没有重叠的。另外，当模型尺寸很小时，仿真结果可能与实际部署后的结果不相符，但我们在此声明这是极少发生的情况。因为对于因通信过程而成为瓶颈的模型而言，其所传输的模型尺寸都相对较大。

5 实验结果

5.1 超额认购场景



a）10 Gbit/s　　b）40 Gbit/s　　c）100 Gbit/s

ICT：每轮迭代过程的完成时间

RAT：新型拓扑感知的参数交换方案

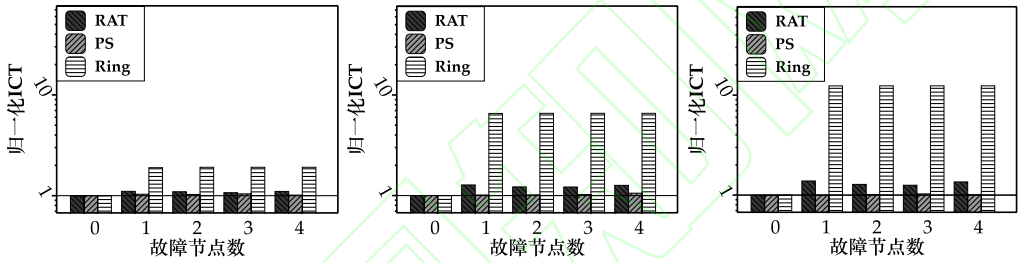
PS：参数服务器

Ring：环形全局规约

图 6 3 种方案在超额认购场景下的仿真结果

如图 6 所示，一方面，PS 在所有不同带宽设置下的平均性能比 RAT 差 25 倍左右。这是因为它引入了大量的跨机架通信流量，从而导致跨机架链路成为瓶颈，影响了任务整体训练速度。另一方面，Ring 将机架间的通信流量最小化，因此我们期望其性能会与 RAT 的结果大致相同。然而，从图中我们发现 Ring 在许多带宽设置下，相较于 RAT，存在 0.16 倍的性能下降。我们通过分析认为，Ring 的长依赖链可能导致在每一跳上都引入一些额外的延迟，这些累积起来的延迟影响了整个训练过程。

5.2 网络故障场景



（a）d=1 ms　　（b） d=5 ms　　（c）d=10 ms

ICT：每轮训练迭代过程的完成时间

RAT：新型拓扑感知的参数交换方案

PS：参数服务器

Ring：环形全局规约

图 7 3 种方案在网络故障场景下的仿真结果

我们还在网络故障的场景下模拟实验，来体现 RAT 对网络不确定性事件的弹性应对能力。如 5.1 所述，我们在给定拓扑中构建一个存在故障问题的网络，并在其上部署了一个分布式机器学习任务。

结果如图 7 所示，与 PS 和 RAT 相比，Ring 在网络存在故障节点的情况下出现了非常严重的性能下降（在最坏情况下平均下降 12 倍），这与我们在第 2 章中的分析一致：如果在 PS 或 RAT 模式下，当存在节点出现故障时，其他正常节点仍然可以利用可用链路带宽继续传输数据。对于 Ring 而言，由于“链阻塞”现象，故障节点的下游节点也全部被阻塞。此外，

由于我们在 Ring 模式下选取故障节点的随机性，可能造成某些节点始终被阻塞的情况——因为过程中可能不仅它本身在某些时刻出现故障被阻塞，而且在其他时间内被其上游的某些节点“链阻塞”。与之对应，RAT 获得了与 PS 相近的性能，这是因为它只引入了相对于 PS 的依赖长度为 1 略长的最小额外依赖长度（在本例中链长为 2）。

6 结束语

本文提出了一种具有拓扑感知能力的新型 DML 参数交换方案：RAT。它利用数据中心网络层数较少的性质，针对物理拓扑特征来建立全局规约树。这些树以其分层模式来构造参数聚合模式，即每个聚合节点在规约阶段聚合其超额认购区域内的全部工作节点的本地梯度，并在广播阶段将更新后的参数或梯度广播回工作节点。与已有的参数交换方案相比，RAT 既实现了最小化跨区域流量的目标，又实现了较短依赖链的目标。

参考文献

[1] NARAYANAN D, HARLAP A, PHANISHAYEE A, et al. PipeDream: generalized pipeline parallelism for DNN training [C]//The 27th ACM Symposium on Operating Systems Principles. Ontario,Canada: SOSP, 2019

[2] HO Q, CIPAR J, CUI H, et al. More effective distributed ml via a stale synchronous parallel parameter server [C]//Conference and Workshop on Neural Information Processing Systems 2013. Nevada, United States: NIPS, 2013

[3] LIN Y J, HAN S, MAO H Z, et al. Deep gradient compression: reducing the communication bandwidth for distributed training [EB/OL].[2020-09-20]. https://arxiv.org/abs/1712.01887

[4] BAI W, CHEN L, CHEN K, et al. Information-agnostic flow scheduling for commodity data centers[C] // NSDI 2015. OKLAND, CA, USA: ASM, 2015

[5] ALIZADEH, YANG S, SHARIF M, et al. pfabric: Minimal near-optimal datacenter transport [C]//SIGCOMM 2013. Hong kong, China: ASM, 2013

[6] CHEN L, CHEN K, BAI W, et al. Scheduling mix-flows in commodity datacenters with Karuna [C]//SIGCOMM 2016. Florianópolis, Brazil: ASM, 2016

[7] CHEN L, LINGUS J, CHEN K, et al. AuTo: scaling deep reinforcement learning for datacenter-scale automatic traffic optimization [C]//SIGCOMM 2018. Budapest, Hungary: ACM, 2018

[8] MOSHARAF C, ION S, et al. Efficient Coflow Scheduling Without Prior Knowledge [C]//SIGCOMM 2015. London, UK: ACM, 2015

[9] ZHANG H, CHEN L, YI B, et al. CODA: toward automaticallyidentifying and scheduling coflows in the dark [C]//SIGCOMM 2016. Florianópolis, Brazil: ACM, 2016

[10] SUSANTO H, JIN H, CHEN K. Stream: decentralized opportunistic inter-coflows scheduling for datacenter networks [C]//IEEE International Conference on Network Protocols (ICNP), 2016. Singapore, Singapore: ICNP, 2016

[11] GREENBER A, HAMILTON J,A MALTZ D, et al. The cost of a cloud: research problems in data center networks [C]//SIGCOMM 2009. Barcelona, Spain: ACM, 2009

[12] Oversubscription and density best practices [EB/OL]. (2015-03-12) [2020-08-10]. https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/storage-networking-solution/net_implementation_white_paper0900aecd800f592f.html

[13] ABADI M, BARHAM P, CHEN J, et al. Tensorflow: a system for large-scale machine learning [C]//OSDI 2016. SAVANNAH, GA, USA: ACM, 2016

[14] JIA Y, SHELHAMER E, DONAHUE J, et al. Caffe: Convolutional Architecture for Fast Feature Embedding [EB/OL].[2020-08-10]. https://dl.acm.org/doi/10.1145/2647868.2654889

[15] CHEN T Q, LI M, LI Y T, et al. MXNet: a flexible and efficient machine learning library for heterogeneous distributed systems [EB/OL]. [2020-08-10].https://arxiv.org/abs/1512.01274

[16] ANDREW G. 2017. Bringing HPC techniques to deep learning [J]. Baidu Research, Tech. Rep, 2017

[17] SERGEEV A, DEL BALSO M. Horovod: fast and easy distributed deep learning in TensorFlow [EB/OL]. [2020-08-10]. https://arxiv.org/abs/1802.05799

[18] MAI L, HONG C, COSTA P. Optimizing network performance in distributed machine learning [C]//HotCloud 2015.SANTA CLARA, CA, USA: USENIX, 2015

[19] KIM J, J DALLY W, ABTS D. Flattened butterfly: a cost-efficient topology for high-radix networks [C]//ISCA 2007. San Diego, CA, USA: ISCA, 2007

[20] GOYAL P, DOLLAR P, GIRSHICK R, et al. Accurate, large minibatch sgd: Training imagenet in 1 hour [EB/OL]. [2020-08-10]. https://arxiv.org/abs/1706.02677

[21] CHO M, FINKLER U, KUNG D. BlueConnect: novel hierarchical all-reduce on multi-tired network for deep learning [C]//SysML 2019. Stanford, CA: SysML, 2019

[22] GENG J, LI D, CHENG Y, et al. HiPS: hierarchical parameter synchronization in large-scale distributed machine learning [C]//Proceedings of the workshop on network meets AI & ML 2018. Budapest Hungary: ACM, 2018

[23] LUO L, NELSON J, CEZE L, et al. Parameter hub: a rack-scale parameter server for distributed deep neural network training [C]//SOCC 2018. Carlsbad, California: ACM, 2018

[24] LUO L, WEST P, NELSON J, et al. PLink: efficient cloud-based training with topology-aware dynamic hierarchical aggregation [C]//SysML 2020. Stanford, CA, USA: SysML, 2020

作者简介

万鑫晨，香港科技大学在读博士；主要研究领域为分布式机器学习系统优化及设计。

胡水海，深圳致星科技有限公司首席科学家；现从事高性能网络系统及人工智能系统的设计工作；曾获 2017 年香港电讯（HKTIIT）优秀博士奖学金；已发表论文 13 篇。

张骏雪，深圳致星科技有限公司首席技术官，香港科技大学在读博士；现从事高性能网络系统及人工智能系统的设计工作；已发表论文 4 篇。