



面向分布式 AI 的智能网卡低延迟 Fabric 技术

Low Latency Fabric Technology of Smart NIC for Distributed AI

熊先奎 /XIONG Xiankui¹, 袁进辉 /YUAN Jinhui², 宋庆春 /SONG Qingchun³

(1. 中兴通讯股份有限公司, 中国 深圳 518057;

2. 北京一流科技有限公司, 中国 北京 100083;

3. 国际高性能计算和人工智能咨询委员会, 美国 森尼韦尔 94085)

(1. ZTE Corporation, Shenzhen 518057, China;

2. Beijing Oneflow Technology Co., Ltd., Beijing 100083, China;

3. HPC-AI Advisory Council, Sunnyvale 94085, USA)

摘要: 系统阐述了在分布式人工智能计算负载中使用低延迟 Fabric 技术出现的问题, 包括虚拟化环境支撑、通信原语抽象化、网络拥塞控制等。针对这些问题, 提出了使用基于可编程智能网卡的解决思路。

关键词: 分布式人工智能; 智能网卡; 远程直接内存访问

Abstract: Some issues of using low latency fabric technology for distributed artificial intelligence workload are systematically discussed, including virtualization environment support, communication primitive abstraction, network congestion control, etc. In view of these problems, a programmable smart NIC solution is proposed.

Keywords: distributed artificial intelligence; smart NIC; remote direct memory access

DOI: 10.12142/ZTETJ.202005005

网络出版地址: <https://kns.cnki.net/kcms/detail/34.1228.TN.20200923.1712.002.html>

网络出版日期: 2020-09-24

收稿日期: 2020-08-15

2012年, AlexNet 神经网络获得 ImageNet 图像识别测试的历史性突破。随后, 基于深度学习的人工智能 (AI) 研究和应用再度爆发^[1-2]。深度学习使用多层卷积或循环神经网络和反向传播算法, 通过标签数据集训练特定模型, 在推理阶段提供语音识别、图像识别等 AI 应用。算法、数据集、算力是本轮 AI 成功腾飞的 3 个要素, 三者缺一不可。其中, 以图形处理单元 (GPU) 多指令多数据流 (MIMD) 计算架构提供的高性能算力训练平台成为深度学习实现工程化的基础。

在机器学习领域^[3], 隐含狄利克雷分布 (LDA) 主题模型对文本表征降维后, 生成聚类统计模型, 以支持互联网实现文本分类、头条推荐等。由大规模逻辑回归 (LR) 生成的预测

模型所支撑的金融违约分析应用等机器学习方法, 也同样需要大规模算力部署。

大规模机器学习本质上是一种特殊的高性能计算 (HPC), 因此, 要实现计算的分布式和并行性, 除算力本身外, 还离不开计算范式的定制化设计和计算通信的网络优化。

最新深度学习模型的训练需要消耗更多的算力。鉴于深度学习模型计算主要是稠密型计算, 业界广泛采用 GPU 等协处理器进行并行加速, 但单个协处理器的算力仍无法满足日益增长的算力需求。通过高速互联技术把更多协处理器连接起来, 能够协同输出大规模算力, 可实现点对点数十至上百吉比特的传输带宽, 例如节点内可以使用外设部件互连标准 (PCIe)

或 NvLink 等技术。当单个节点仍无法满足需求时, 可通过高速网络实现多节点分布式计算。此时, 由于传输带宽过低, 普通以太网网络 (千兆网或万兆网) 会出现多节点扩展效率极低、计算资源严重浪费的现象。因此, 在分布式深度学习训练场景中, 基于远程直接内存访问 (RDMA) 的网络通信成为最佳选择。

分布式深度学习训练任务必须使用 RDMA 技术, 这是因为: (1) 深度学习训练任务普遍使用随机梯度下降算法。每处理一小片数据就需要更新模型参数, 计算粒度很细, 对网络传输的延迟容忍度非常低。(2) 深度学习普遍使用的 GPU 加速卡, 吞吐率非常高。如果数据搬运速度跟不上计算速度, 就容易造成计算资源浪费。

(3) 在深度学习训练中，系统调度、数据加载和预处理均需要使用 CPU 资源，而基于传统以太网的网络传输也需要消耗很多 CPU 计算资源，这会影 响整个系统的效率。RDMA 的内核旁路技术可降低 CPU 利用率，提高整个系统的效率。

因此，分布式 AI 训练平台需要特殊的硬件设计、低延迟的通信网络、抽象化的通信原语库和定制化的计算范式。其中，低延迟的通信网络和通信原语库是本文关注的重点。

1 RDMA 低延迟特性和技术背景

RDMA 技术是由无限带宽贸易协会 (IBTA) 定义的面向高性能、大规模、软件定义及易管理等的先进网络通信技术^[4]，最初被用于 InfiniBand 网络，后来被广泛应用在以太网中。基于融合以太网的 RDMA (RoCE) 更是被全球各大数据中心用户广泛地部署。RDMA 已成为目前提升网络应用性能的关键技术之一。

RDMA 技术提供了一个可以将 RDMA 硬件对外进行高效通信的抽象层 (verbs)。用户可以利用各种方式来调用 verbs，以实现对于 RDMA 硬件的调用^[5]。一般而言，虽然直接调用 verbs 可以获得最佳性能，但是仍然需要对应用的通信接口进行修改。当然，也可以直接使用现有的通信中间件来调用 RDMA，而无须对应用做任何修改，如信息传递接口 (MPI)^[6]、英伟达多 GPU 通信库 (NCCL)^[7]、Internet 小型计算机系统接口 (iSCSI)、基于 Fabric 的非易失性存储 (NVMeoF)、Lustre 等。这是因为这些中间件已经将 RDMA 通信库包含在自己的通信库里，对应用而言是完全透明的。

在通信过程中，RDMA 可以绕开操作系统，直接实现不同机器内存之间的数据传递。如图 1 所示，与传输

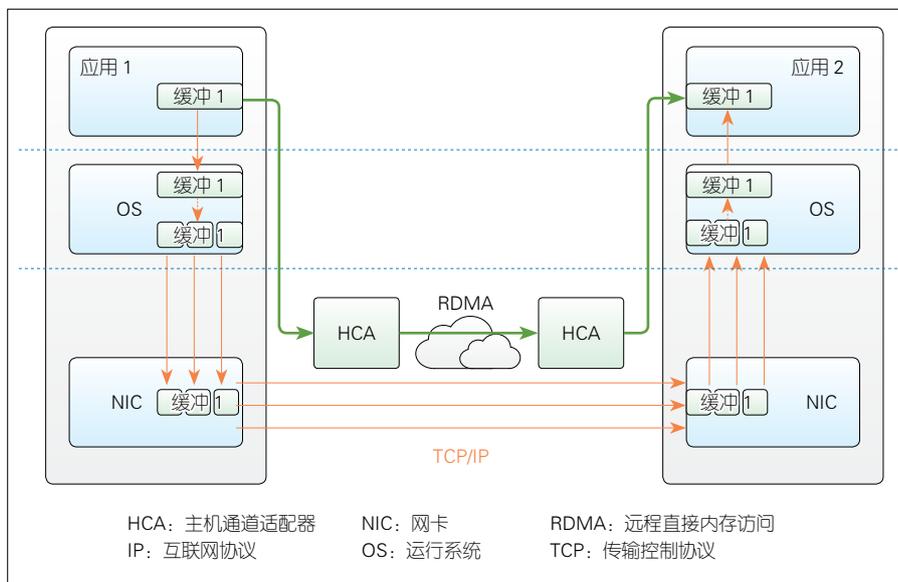
控制协议 (TCP) 相比，RDMA 不需要从内存到操作系统、再到网卡硬件的层层拷贝，也不需要等待 CPU 中断来层层拷贝数据。这既降低了 CPU 在通信上的利用率，又减少了对 CPU 中断的消耗。将 CPU 或者 GPU 的计算资源完全供给应用有助于应用性能的提升。此外，随着网络速度的提升，传统的 TCP 通信对 CPU 在通信上消耗的资源也越来越多，这导致应用得不到足够的 CPU 资源来运行。当数据中心规模达到一定程度时，可能会出现性能下降的现象，这也是为什么大规模数据中心都最先关注 RDMA 的原因之一。

在通信中，RDMA 是如何做到不消耗 CPU 呢？在网络传输层，IBTA 规范首先做了严格的定义，即将传输层的各种操作都直接定义到了硬件实现上，例如将应用的 message 在发送端传输时的分割打包操作。由于应用的信息往往都很大 (最大达到 2 GB)，网络无法将一个这么大的信息直接传出去，需要将它分割成多个小于网络最大传输单元 (MTU) 的数据包，然后给每个数据包做 Checksum 和加 header

等。所有这些操作原来都是 CPU 在做，现在 RDMA 网卡可以直接将它们卸载过来，无须再依赖于 CPU。同样地，在接收端也可以将一系列反向操作以卸载到网卡的硬件上来。此外，IBTA 还根据网络面向数据或是面向连接等特性定义了不同的数据传输方式，如可靠连接 (RC)、不可靠连接 (UC) 和不可靠数据报 (UD) 等。利用各种数据传输方式的特点，应用之间可以使用不同的 RDMA 通信方式，如 rdma_send、rdma_write、rdma_read、rdma_atomic 等，如图 2 所示。

rdma_send 是一种双边操作，需要在发送端和接收端都执行后处理请求操作。当数据到达接收端时，需要向远端服务器查询数据的终点，这时可以使用 RC、UC 和 UD 来进行数据传输。

rdma_write、rdma_read 和 rdma_atomic 都是单边操作。使用 rdma_write 时，远端 CPU 在数据写入的过程中不参与任何操作，可以使用 RC 或 UC 来传输；使用 rdma_read 时，在从远端存储中读取数据的过程中不需要通知远端 CPU，可以使用 RC 来传输；rdma_



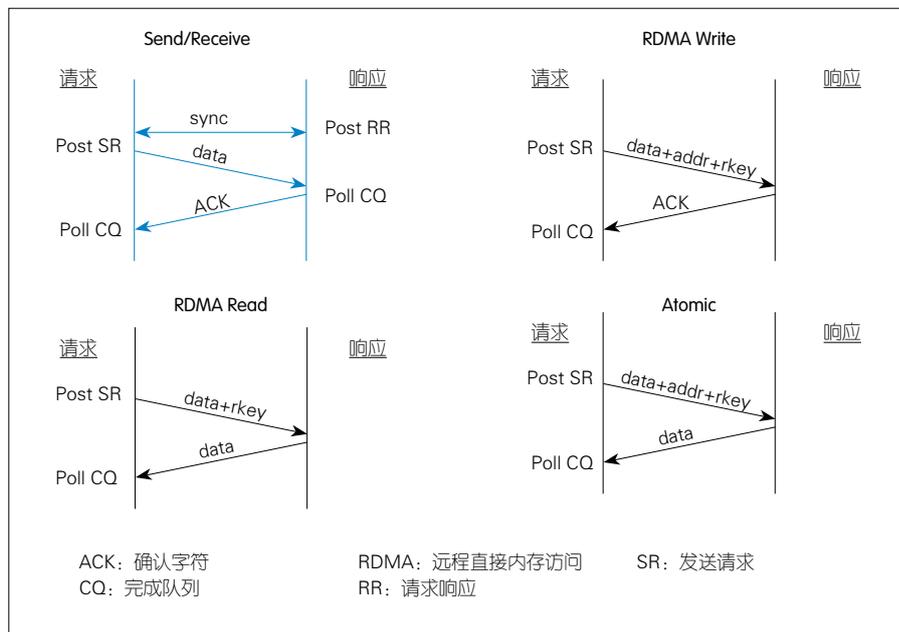
▲图 1 RDMA 和 TCP 的比较

atomic 是 rdma_read 和 rdma_write 的组合操作，应用可以首先从远端服务器中读取数据、改写数据，然后再写回远端内存，中间不需要和远端 CPU 有任何沟通，可以使用 RC 来传输。

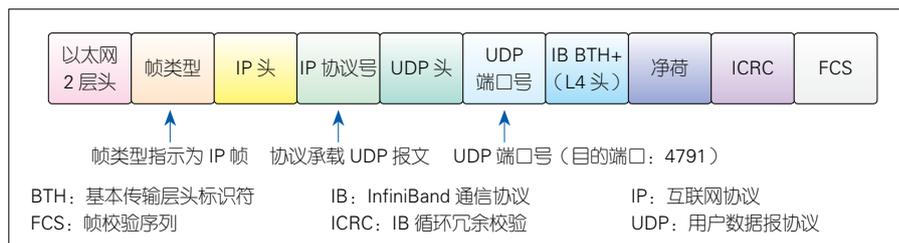
通过在传输层的卸载，RDMA 实现了应用之间的高效通信，但这同时给 RDMA 网络的端到端连接带来了挑战：如何能保障在数据传输的过程中不会因网络问题而导致数据的丢失。为了应对这一挑战，在 InfiniBand 网络中，采用了基于 credit 的链路层流控机制。高性能的动态路由和拥塞控制机制有效地控制了网络中的丢包问题。在以太网上，显示拥塞通知 (ECN) + 拥塞通知包 (CNP) 和基于优先级的流量控制 (PFC) 等机制被用于减少网络中的丢包，但如何将以太网的流控做到像 InfiniBand 网络那样高效和可靠，还需要很长的路要走。

目前，基于以太网的 RDMA 主要标准为 RoCE，包括 RoCE V1 和 RoCE V2 两个版本。其中，RoCE V1 主要是面向同一个以太网子网内的 RDMA 通信，使用较少，只在早些时候的计算和存储应用中使用过；RoCE V2 实现了 RDMA 协议在以太网上的跨子网通信，可以进行大规模部署和远距离传输。目前主流的 RoCE 应用都是基于 RoCE V2 的，并且都是先将 InfiniBand 的完整数据包作为 RoCE 的 Payload，然后再通过用户数据报协议 (UDP) header 和传统的以太网 header 对接，实现 RDMA 包在传统以太网网络上的传输。

图 3 给出了 RoCE 的数据格式。RoCE 作为一种高效的以太网通信方式，可以应用在不同场景：(1) 在无损 (Lossless) 以太网络中，需要使能 ECN、PFC 和端到端服务质量 (QoS)，需要的配置虽然相对复杂，但是有助于提升网络性能。(2) 在有损 (Lossy) 以太网络中，不使能 PFC，可选使能



▲图 2 RDMA 传输方式举例



▲图 3 基于融合以太网的远程直接内存访问数据格式

ECN，配置简单，但需要更先进的数据完整性控制机制^[8-9]。

RDMA 和 RoCE 技术已经被广泛地应用于 HPC、存储、深度学习、机器学习、数据库、大数据等应用场合中。与此同时，RDMA 技术也为未来技术的发展带来更加广阔的研究空间，成为现代以数据为中心的系统架构的核心。

2 分布式 AI 训练平台的定制化硬件

如图 4 所示，分布式 AI 训练服务器采用定制化硬件设计。其中每个服务器节点由 2 个 XEON CPU 与 8 块 GPU 模块提供主要算力。该系统被分割成两个通信域：节点内部通信和分布式节点间通信：

(1) 节点内部通信。GPU 间采用 NVlink 构成超立方体拓扑连接，可通过 GPU Direct 共享内存技术和 NCCL 通信原语，支持模型数据和暂存数据在 HBM 内存中进行低延迟交换和传递。GPU 和 CPU 间、GPU 和网卡 (NIC) 间可以通过 PCIe 对等网络 (P2P) 通信支持数据的直接内存访问 (DMA) 搬移和共享。

(2) 节点间通信。AI 训练平台由大量的 GPU 服务器节点组成的分布式系统构成。分布式节点间通过 RDMA 低延迟 Fabric 网络连接。分布式系统通信方式可以是 MPI、NCCL 或者远程过程调用 (RPC) 通信，但一般采用效率更高的基于 RDMA 原生 Verbs 的通信原语库。

3 分布式 AI 计算范式

分布式 AI 计算范式从计算类型上大致可分为两种类型：模型并行和数据并行。

(1) 模型并行。当模型比较大时，GPU 自带的 HBM 高带宽内存难以全部容纳放入，这时可以把模型数据按神经网络层次或者参数分区进行拆分，并分布到不同的 GPU 上进行独立计算，然后再统一汇聚进行迭代。

(2) 数据并行。不同 GPU 分配同样的模型参数，并将待训练的迭代数据分批次加载到不同 GPU 进行训练。不同批次数据训练结果以某种规则合并迭代后完成训练。

模型并行需要拆分模型参数，因此，GPU 间存在着较强的计算图依赖性，往往难以弹性伸缩；而数据并行则要好得多，更适合大规模的分布式 AI 训练。

在实际应用时，往往采用数据并行 + 模型并行的混合模式。大型模型参数可以先按照模型并行拆分形成独立计算簇，再通过数据并行模型对计算簇进行并行化扩展。

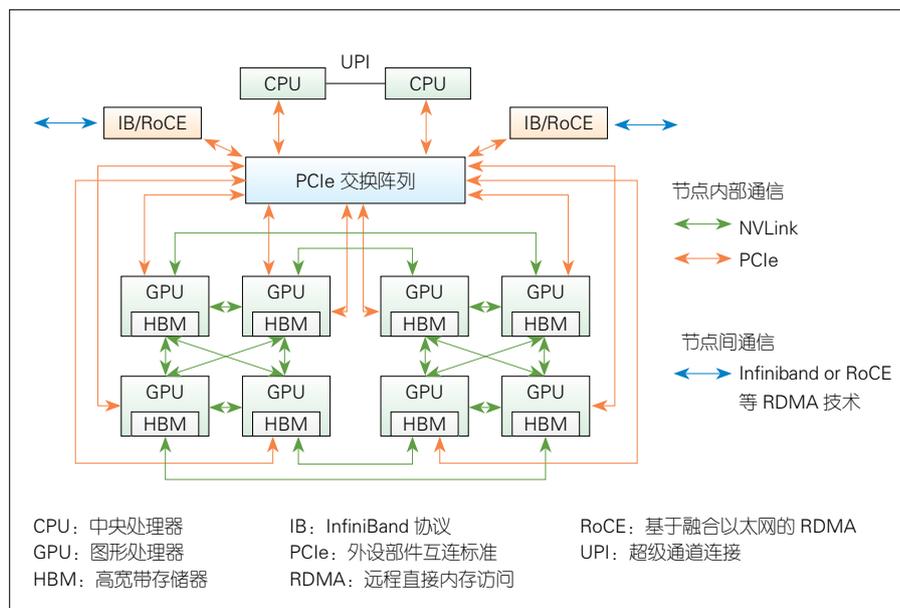
在并行化方面，分布式 AI 平台也存在两种主流类型：参数服务器(PS)模型和环形拓扑模型。

如图 5 所示，PS 模型是当前重要的分布式 AI 训练平台计算范式。本质上，PS 是一种类似于 Hadoop 分布式文件系统的半集中计算架构，采用了类似 Map-Reduce 的计算范式。其中，PS 服务器和控制器起到中央调度器的作用，各个 GPU 节点充当任务对象并负责计算各自的梯度更新。PS 服务器负载 Broadcast 模型参数给所有任务对象，并负责在 PS 服务器上汇聚所有梯度更新，以完成一个批次数据的迭代。显而易见，PS 服务器是整个系统的瓶颈点，并且高度依赖通信网络的低延迟和高带宽。

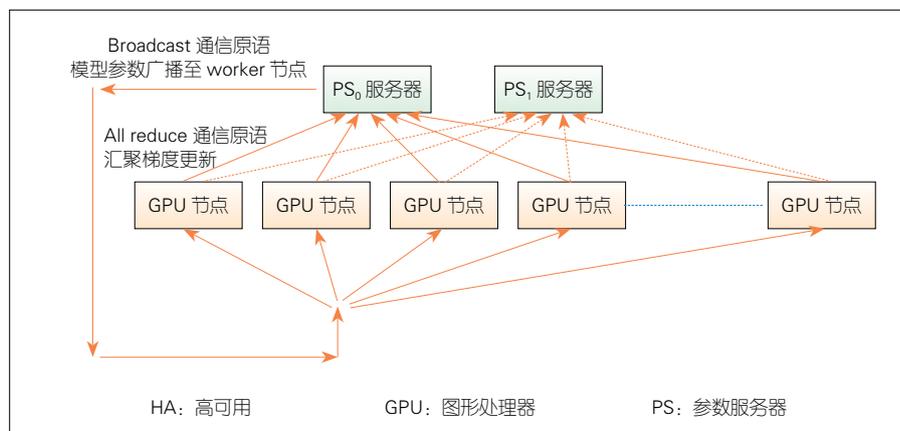
如图 6 所示，环形分布式计算范式具有 GPU 节点逻辑成环的特点，没有 PS 计算范式的聚合服务器半集中点。每个 GPU 任务对象接收上个节点批次计算产生的梯度数据更新，并计算本节点的梯度再传递给下一个任务对象节点。从调度上看，环形计算充分利用了深度学习训练过程反向传播(BP)链式求导从后向前过程的特点：后级网络层先计算，逐步推到前级网络层。因此，梯度计算和梯度传递交替进行，这充分利用了通信带宽和计算流水性。环形计算范式一般使用 Reduce-Scatter 和 All-Gather MPI 通

信过程。

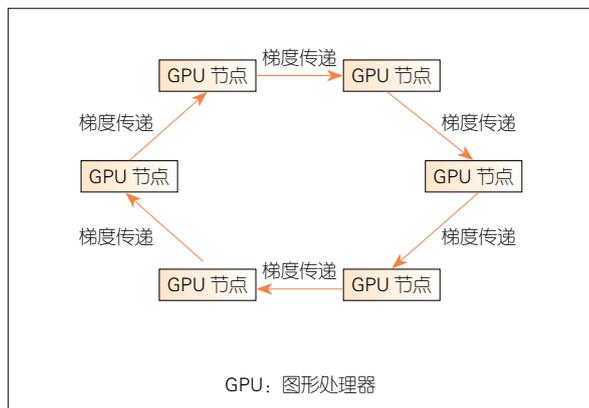
在分布式并行计算范式中，还同时存在着同步通信和异步通信的差异。传统分布式系统，例如大数据排序(Sorting 操作)，大多采用块同步并行模型(BSP)同步通信，即在一次迭代中设置 Barrier 以等待系统最慢的节点完成此次计算。然而在神经网络和机器学习中，模型参数的更新具有一定鲁棒性，因此，可以接受延迟同步并行计算模型(SSP)异步方式更新模型数据。在 SSP 模式下，允许计算快的节点领先最慢的节点若干个迭代步长，而不是仅仅等待慢节点追赶进度。



▲图 4 GPU 服务器节点通信框图



▲图 5 参数服务器计算范式



▲图 6 环形分布式计算范式

4 分布式 AI 中使用低延迟 Fabric 的一些问题

4.1 RDMA 固有问题

如前文所述，RDMA 是建立在旁路内核、通过硬件维护内存区和队列管理获取低延迟的低延迟 Fabric 技术，因此，RDMA 会不可避免地存在以下问题：

(1) 硬件低级别的通信原语过于抽象，导致现代数据中心应用软件难以适配。一方面，低级别的通信库和协议栈使得应用软件需要关注硬件级别的 buffer 管理问题，因此，一般都需要在原生 RDMA 驱动和库上进行一层封装抽象，这导致 RDMA 性能不能充分发挥；另一方面，在现代数据中心云基础设施里，网络普遍通过软件定义网络（SDN）Overlay 网络和 vSwitch 框架实现虚拟化隔离和租户安全，而原生 RDMA 技术在穿越隧道和分段网络时，难以维持缓冲和队列管理。因此，业内多妥协采用裸金属物理服务器和原生网络部署以换取原生 RDMA 性能。

(2) 目前，RDMA 技术缓冲资源管理是在网卡专用集成电路（ASIC）芯片内实现的，用户态应用旁路内核直接操作网卡硬件。从商业角度看，CPU 和 NIC 芯片分离、互相解耦有利

于建立生态系统，然而，无论是在资源保护和硬件性能可扩展性上，还是在应用间共享资源（零拷贝技术、fd.IO 包向量处理技术）等方面，ASIC 化的网卡都欠缺灵活性。

(3) 基于以太网的 RDMA 技术有 RoCE 和 iWarp，其中比较流行的是 RoCE（V2 版本）。如前

文所述，在有损或无损以太网环境下，由于以太网 Best-effort 问题和组网部署特点，RoCE 网络存在着规模扩展性问题^[10]。采用 PFC 流控技术显得过于“粗暴”，并且在 In-Cast 和拥塞环境下，节点数目很难扩展到实用级别规模。采用数据中心量化拥塞通知（DCQCN）和 TIMELY（Delay）等技术方案仍不能解决参数收敛慢、全系统调优困难等规模扩展性问题。关于这一点，阿里巴巴高精度拥塞控制（HPCC）算法已给出详细论述^[11]，此处不再赘述。

4.2 分布式 AI 中使用 RDMA 面临的问题

现有深度学习框架极少直接基于 RDMA 技术开发网络通信库，通常都是通过 Nvidia NCCL 来间接使用 RDMA，例如 TensorFlow、PyTorch、PaddlePaddle、MxNet 等主流深度学习框架。这可能导致两个潜在的问题：一方面，非 Nvidia 通用图形处理器（GPGPU）的加速器厂商如果想要实现分布式深度学习训练功能，就必须自己开发类似于 Nvidia NCCL 的通信库来使用 RDMA；另一方面，深度学习框架与底层通信库相隔离，无法进行一体优化^[12-13]。

对于基于 RDMA 技术实现底层网络通信，最主要的挑战是内存管理。RDMA 传输需要使用注册内存（锁页

内存），如果每一次网络传输都根据传输量的实际需求当场申请并注册，则会导致开销显著，还会增加网络传输的延迟。如果提前申请并注册好内存块，则可能因无法准确预知程序运行时每次的实际传输需求而过多申请内存，这会造成内存资源的浪费。

针对这一问题，一般有如下几个解决方法：（1）提前申请并注册大块内存，在实际传输时使用内存池技术，并从已注册好的大块内存上分配需要的内存；（2）提前申请并注册一些固定大小的内存块，每次在进行数据传输时，首先把需要传输的数据拷贝到这些已注册内存块上，然后再通过 RDMA 进行传输；（3）对于静态结构的神经网络，每一次迭代网络传输量是固定不变且可提前预知的，在系统运行前申请并注册内存，然后重复使用这些注册内存块。

OneFlow 软件是目前唯一一个内部原生集成 RDMA 网络传输功能的深度学习框架。用户既可通过 Nvidia NCCL 使用 RDMA 传输，也可直接使用 OneFlow 基于 RDMA 自研网络通信库去实现分布式训练。针对动态形状和静态形状的网络传输，可分别使用上文描述的第 2、3 种技术解决方案。动态形状的网络传输须引入一次额外的拷贝，而静态形状的网络传输则实现了真正的零拷贝，达到了同时兼顾内存利用率和运行效率需求的目的。

OneFlow 软件还充分发挥了 RDMA 的优点，采用 Actor 软件机制实现了一个简洁的去中心化调度系统框架。该软件先在编译阶段生成静态计算图，然后生成包含 Actor 实例的分布式环境描述信息的计划。分布式系统最终根据计划生成 Actor 实例运行态（各 Actor 间的生产、消费数据会被存储在 Register 中），通过 RDMA 低延迟网络消息传递协作来完成计算流水。

5 中兴通讯对低延迟 Fabric 技术的探索

中兴通讯在智能网卡领域有较强的技术积累, 相关技术已被广泛使用在 5G 用户面功能 (UPF) 下沉实现的低延迟 2B 业务、5G 核心网 L3 虚拟专用网络 (VPN) IPsec 加解密连接、数据中心输入输出 (IO) 虚拟化等各种产品应用和方案中。

智能网卡实现低延迟 Fabric 的优势包括: (1) 比商用网卡更为强大的性能和资源保证, 如更大的最长前缀匹配 (LPM) 精确匹配查找表和更快的三态内容寻址存储器 (TCAM) 模糊匹配资源; (2) 高度灵活性和可编程性, 即根据应用需要调整数据流和资源抽象化; (3) 软硬协同一体化设计, 如可与分布式 AI 框架协同优化甚至定制; (4) 模块化, 即利用已有模块实现快速的系统集成; (5) 基于现场可编程门阵列 (FPGA) 实现智能网卡, 以便于未来实现批量芯片化。

基于智能网卡技术, 中兴通讯做了一些低延迟 Fabric 框架上的技术探索, 以尝试解决前文所述相关问题。首先, 将 RDMA 网卡设备原型下沉进入 FPGA, 并基于 VirtIO 前后端技术实现了 IO 虚拟化。将 VirtIO 后端和 vSwitch、Vxlan 隧道端点 (VTEP) 卸载至 FPGA 中, 以适配数据中心虚拟化网络支持问题。其次, 基于 TCP 减负引擎 (ToE), 将 TCP 状态机下沉进入 FPGA, 以支持低延迟用户态 TCP 协议栈 (基于 FStack 原型)。此外, 通过通信原语库封装抽象和智能网卡硬件先入先出 (FIFO) 桥接技术, 对底层 RDMA 通信原语进行虚拟 Socket 封装, 可对应用实现近似伯克利套接字 (BSD Socket) 的封装调用。

针对低延迟 Fabric 框架技术, 中兴通讯下一步的研究方向包括:

(1) 基于 P4 框架实现网卡拥塞

控制相关算法, 通过增加带内网络遥测 (INT) 的遥测能力来减少对往返时延 (RTT) 和确认字符 (ACK) 丢失的依赖, 并尝试通过控制面获取网络转发设备的 Buffer 水位信息, 以获得更精确的调度能力;

(2) 除已支持对通信原语库中的广播机制进行硬件加速外, 后续将在通信原语库抽象化封装中增加实现应用层感知接口, 支持可配置选项 (如带宽、时延、突发、抖动、拥塞优先级等参数) 提供给网卡设备层以配合控制算法实现网络规模扩展;

(3) 针对有损环境和偶发 In-Cast 丢包环境, 通过采用类似低密度奇偶校验码 (LDPC)、前向纠错码 (FEC) 等交叉交织冗余编码算法, 减少和平滑重传所导致的网络延迟问题;

(4) 针对边缘计算区块链、联邦学习等广域分布式 AI 应用, 对同态加密等强计算需求进行加速。

6 结束语

由于受限于智能网卡 FPGA 资源, 低延迟 Fabric 技术的相关探索仍停留在原型阶段, 离商用化还有不小的距离。谨以此文抛砖引玉, 共同探索低延迟 Fabric 技术的落地。

参考文献

- [1] 王海宁. 网络人工智能发展分析与建议 [J]. 中兴通讯技术, 2019, 25(2): 47-51. DOI: 10.12142/ZTETJ.201902008
- [2] 张嗣宏, 左罗. 基于人工智能的网络智能化发展探讨 [J]. 中兴通讯技术, 2019, 25(2): 52-56. DOI: 10.12142/ZTETJ.201902009
- [3] XUE S Y, LI A, WANG J F, et al. To learn or not to learn: deep learning assisted wireless modem design [J]. ZTE communications, 2019, 17(4): 3-11. DOI: 10.12142/ZTECOM.201904002
- [4] InfiniBand. VMware: InfiniBand and RDMA better than ethernet [EB/OL]. [2020-09-10] <https://www.infinibandta.org/>
- [5] Open MPI. Open source high performance computing [EB/OL]. [2020-09-10]. <https://www.open-mpi.org/>
- [6] Open Fabrics Alliance. OFED for Linux [EB/OL]. [2020-09-10]. <https://www.openfabrics.org/ofed-for-linux/>
- [7] NVIDIA Developer. NVIDIA NCCL [EB/OL].

- [2020-09-10]. <https://developer.nvidia.com/nccl>
- [8] Mellanox Technologies. How to enable/disable lossy roCE accelerations [EB/OL]. [2020-05-24][2020-09-10] <https://community.mellanox.com/s/article/How-to-Enable-Disable-Lossy-RoCE-Accelerations>
- [9] Mellanox Technologies. Recommended network configuration examples for RoCE deployment [EB/OL]. [2020-09-09][2020-09-10] <https://community.mellanox.com/s/article/recommended-network-configuration-examples-for-roce-deployment>
- [10] ZHU Y B, ZHANG M, ERAN H, et al. Congestion control for large-scale RDMA deployments [EB/OL]. [2015-08-15][2020-09-10]. <http://conferences.sigcomm.org/sigcomm/2015/pdf/papers/p523.pdf>. DOI: 10.1145/2785956.2787484
- [11] LexiNexis. HPCC Systems: Introduction to HPCC [EB/OL]. [2011-05][2020-09-10]. https://cdn.hpccsystems.com/whitepapers/wp_introduction_HPCC.pdf
- [12] Oneflow-Inc. Ibverbs [EB/OL]. [2020-09-10]. https://github.com/Oneflow-Inc/oneflow/tree/master/oneflow/core/comm_network/ibverbs
- [13] Oneflow-Inc. Verbs [EB/OL]. [2020-09-10]. https://github.com/tensorflow/networking/tree/master/tensorflow_networking/verbs

作者简介



合研究工作。

熊先奎, 中兴通讯股份有限公司无线首席架构师、中兴通讯股份有限公司计算存储技术委员会主任、移动网络和移动多媒体技术国家重点实验室学术委员会委员; 曾主导开发 ATCA 电信计算平台和服务器存储平台; 主要从事 ICT 融



1%), 曾连续多年获美国国家技术标准局组织的视频检索评测比赛第一名; 发表论文多篇。

袁进辉, 北京一流科技有限公司创始人、之江实验室天枢开源开放人工智能平台架构师; 致力于打造新一代深度学习框架 OneFlow; 在微软任职时曾发明世界上最快的主题模型训练算法 LightLDA, 获微软亚洲研究院院长特别奖 (Top



宋庆春, 国际高性能计算和人工智能咨询委员会亚太区主席、亚太区 HPC-AI 竞赛和 RDMA 编程竞赛主要负责人; 致力于 RDMA 编程技术在亚太区的教育和推广, 拥有超过 20 年的 HPC、存储和高性能网络经验。