阿里巴巴 EFLOPS 集群系统：大规模AI实践孵化的算力基础设施

曹 政 董建波 金铃铃 等
阿里云计算有限公司

关键词：深度学习 智能计算 集群 软硬件一体化

引言

深度神经网络在机器学习领域的突破性进展，推动社会经济生活步入了智能的大计算时代，算力需求呈现了爆炸式增长。然而，智能计算业务的计算模型和通信模型，与大数据处理 / 高性能计算业务存在显著差异，传统基础设施无法为高速发展的智能计算业务提供有效的算力保障 [1]。因此，AI 集群架构设计亟须结合智能计算业务的特征，通过软硬件一体化设计，实现智能算力从基础设施到业务算法的无损传递。本文将重点介绍阿里巴巴的 EFLOPS AI 计算集群硬件设计，它孵化自阿里巴巴丰富多样的业务实践，在提供高效算力的同时兼备应用普适性。

智能计算的背景和挑战

目前，人工智能已经渗透到教育、交通、能源、气象等众多传统行业，在互联网领域更是得到了广泛的应用。以淘宝为例，该网站的商品搜索、智能

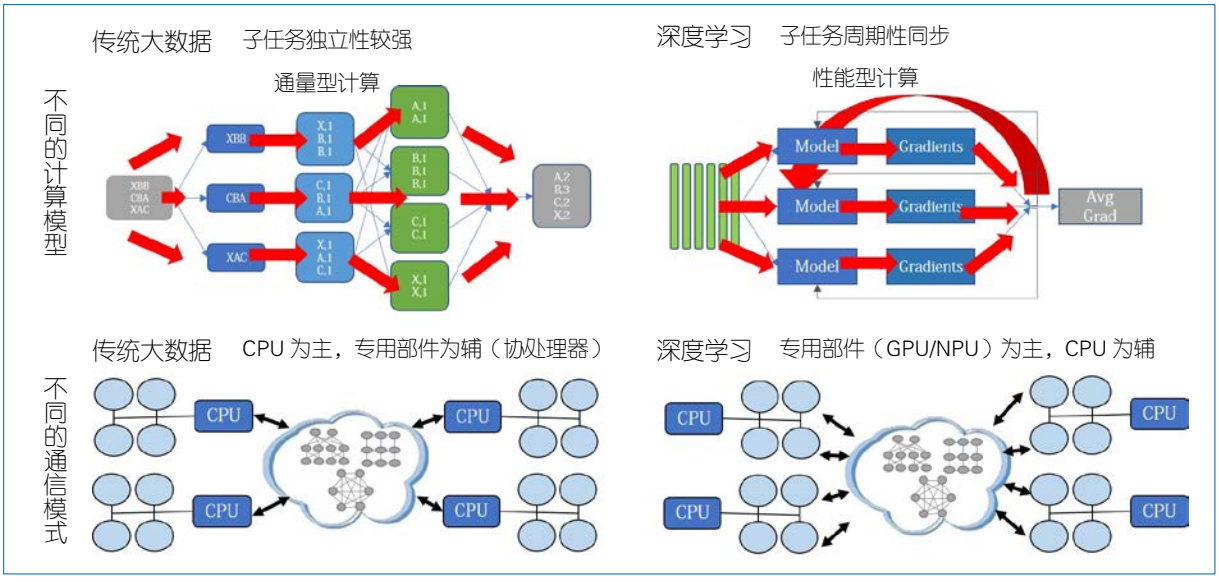

图 1 深度学习与传统大数据应用的差异

客服、个性化推荐等功能，每天涉及 10 亿张图片、120 万小时视频、55 万小时语音和 5000 亿句自然语言处理，背后都是以大数据和 AI 为代表的智能计算在支撑，全面覆盖了视觉、自然语言处理和点击率预估模型（Click-Through Rate，CTR）等典型人工智能应用领域。

随着神经网络在图像语音识别等领域的成功应用，以此为基础的深度学习算法也迅猛发展。为了获得更好的识别准确率，神经网络的模型和相关的训练数据集急剧增长，复杂大模型训练的时间也开始以“星期”甚至“月”为单位计量，阿里巴巴的众多 AI 业务模型开始进入大模型时代。例如，拍立淘增量训练的数据集一般为 2 亿张图片，使用一个英伟达 Volta 100 GPU（峰值 Tensorcore，算力 112 TFlops）进行训练需要一周时间；而从零开始训练需要 10 亿张图片，需要 1.5 个月才能完成，这种训练速度远远满足不了业务快速迭代的需求。为了缩短训练时间，构建分布式 AI 训练集群以实现系统性能的横向扩展成为必然选择，谷歌的 TPU 集群 [2] 就是类似系统。

由于深度学习与大数据 / 高性能业务有着不同的计算模型和通信模型，因此采用传统的集群架构将面临严重的通信瓶颈，导致系统扩展性急剧下降。如图 1 所示，传统大数据处理业务，如 Hadoop 任务，其子任务独立性强，处理流程以流式或单向图为主。整体系统呈现通量计算特征，即以单位时间完成的任务数为重要指标。而 AI 业务则是高性能计算负载，下一次计算任务的启动依赖于上一次计算的完成，因此其重要指标是单任务的完成时间，需要极致的计算性能，因此高性能的加速器（如 GPU）被广泛应用在 AI 系统中。而加速器的规模使用，给通信模型带来了新的挑战，传统的通信模式从原有的 CPU 间通信转向 CPU 与加速器间通信。如果仍然沿用原有以 CPU 为中心的通信架构，势必会引入复杂的通信层次和数据拷贝，造成极低的通信效率。随着加速器性能的不断提升，计算部分时间占比不断缩短，进而使得通信的瓶颈问题愈加凸显，极大制约了系统性能的线性扩展性能。图 2 是 BERT 模型训练性能的曲线，可以看到应用扩展性在低速网络和高速网络下有巨大的剪刀差。

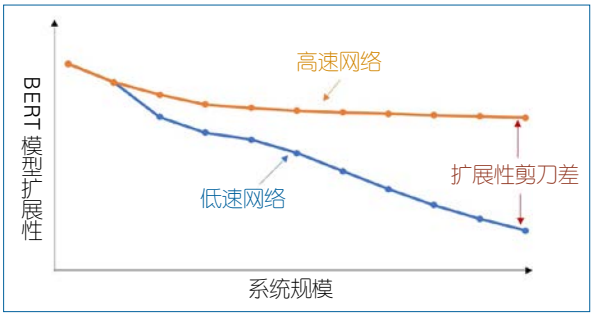

图 2　BERT 模型训练网络性能比

因此，算力绝不是一个简单的计算芯片堆叠，而是突破服务器界限在集群尺度开展计算、网络、存储协同设计的结果，而最佳的协同设计则依赖对 AI 真实负载特征的充分理解。真实的 AI 计算任务包含数据集的读入 / 预处理、模型计算、梯度同步等多个环节，全面涉及集群系统的数据 IO、通用 / 异构计算、网络通信等组件。实际上，我们观察到大量 AI 业务的运行效率受限于通信和数据 IO 性能，而非计算，因此单纯增加异构计算部件的性能对系统整体性能的帮助不大。要全面提升 AI 业务执行效率，首先需要对 AI 业务有充分的理解，在硬件层面平衡系统通用 / 异构计算资源的配比，协调计算 /IO/ 通信的性能，与软件系统深度协同设计，才能充分发挥算力资源的效率。

典型 AI 负载场景

阿里巴巴多样性的业务基本实现了典型场景的全覆盖。根据计算特征，我们可以将 AI 场景简单地分为稠密计算和稀疏计算两大类。其中视觉和自然语言处理属于稠密计算的范畴，而推荐广告类模型属于稀疏计算的范畴。

顾名思义，稠密计算的模型输入是稠密的张量，每个元素都参与模型的计算。一般情况下，采用数据并行方式将模型复制到多个加速部件，并发处理多组数据（超大模型也将导致模型切分和相应的模型并行计算）。各个加速部件完成梯度的计算后，通

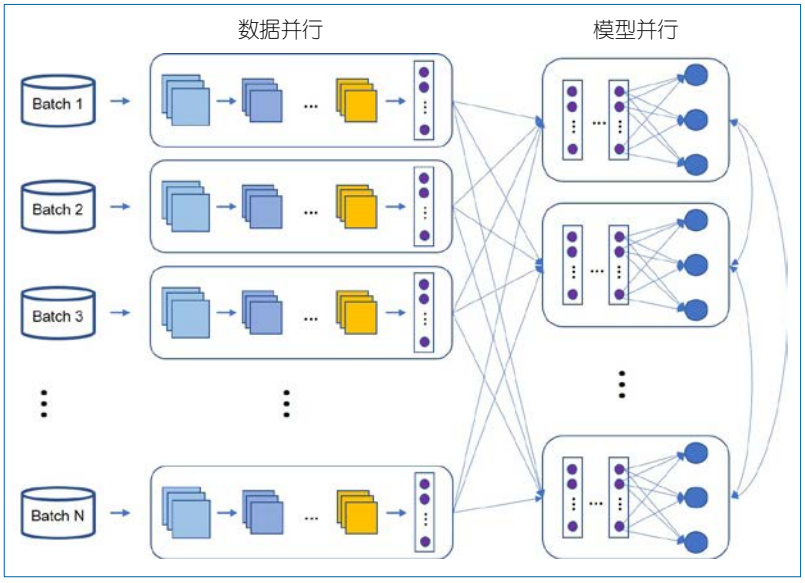

图 3　数据与模型混合并行

过 AllReduce 等集合通信操作实现梯度的全局同步和模型参数的更新。对稠密模型而言，梯度的全部同步通信开销会限制 AI 计算集群的扩展性。

稀疏计算的模型往往具有非常大的特征空间，而每个批处理（batch）的输入中仅包含少量有效特征。稀疏模型的计算一般包含稀疏特征的嵌入（embedding）和后面的稠密计算两个部分。其中，稀疏特征的嵌入计算是其显著区别于稠密计算的关键部分。训练中模型需要对特征嵌入表（embedding table）进行大量查找、排列等操作，然后才能对生成的张量做计算（如 attention 或者 MLP）。极具稀疏特征的嵌入表通常占用非常大的存储空间，无法在加速部件中完整存放（甚至无法在单个节点上完整存放），因此稀疏计算往往采用模型并行的方式，将整个模型切分到多个加速部件甚至服务器。这就导致了任意一组数据的计算都需要从多个服务器获取模型参数，因而形成全网状（Full-Mesh）网络通信模型。比如，在 PS（Parameter Server，参数服务器）架构中，每个工作服务器（worker）都需要向所有的参数服务器发起参数访问请求。大量并发 Incast（多打一）流量会导致严重的参数访问长尾问题，限制 AI 计算集群的扩展性。

所以，稠密和稀疏计算在业务特征上存在显著的差异，对计算 / 存储 / 通信资源的需求也存在明显的不同。要使昂贵的计算资源效率最大化，需要结合业务特征对 AI 集群的软硬件系统进行合理设计。

阿里巴巴手淘业务的拍立淘功能是稠密计算的应用场景之一，该功能通过图像处理进行大规模货架商品 SKU（Stock Keeping Unit，最小存货单位）识别。在需要识别百万商品时，模型参数大概为 20 亿，需要 6000 万张图片训练；需要识别千万商品时，模型参数高达 50 亿，数据量扩展到 6 亿张图片。由于模型变得过于复杂，单个 GPU 显存已经无法存储所有的参数，因此无法应用传统的数据并行模式，必须采用混合模型并行，如图 3 所示，将模型切开放置到不同的 GPU 上，大量原本在单 GPU 上的数据交换变成了 GPU 间的数据交换，对通信提出了巨大的挑战。评测表明，在 50 Gb/s 的网络下，通信占比超过了 50%，大量时间花费在了通信上。自然语言处理类的应用模型往往规模更大，因而在通信方面面临更大的挑战，比如高达 1700 亿参数的 GPT-3 模型。

阿里巴巴的主搜功能和广告业务大量采用了稀疏计算模型。相比上述的稠密场景，稀疏计算场景的挑战更大，因为稠密模型的资源需求更为复合，对计算资源的平衡、存储空间的容量与分布、计算部件之间的高速通信都提出了较高的要求。一个典型的模型如图 4 所示。在搜索广告场景，千万商品的各个属性、数亿用户的各个属性，连同行为序列被记录成一个超大的向量，其大小可达到百亿量级。存储、通信、计算资源的开销随用户行为序列长度相应地线性增长。由于这个模型参数分布于多个 PS 节点，同步性并发查询导致通信长尾明显，通信占比高（大于 50%），GPU 利用率呈峰谷特性，线性扩展能力极为受限，百节点规模的性能就已经有非常显著的下降。

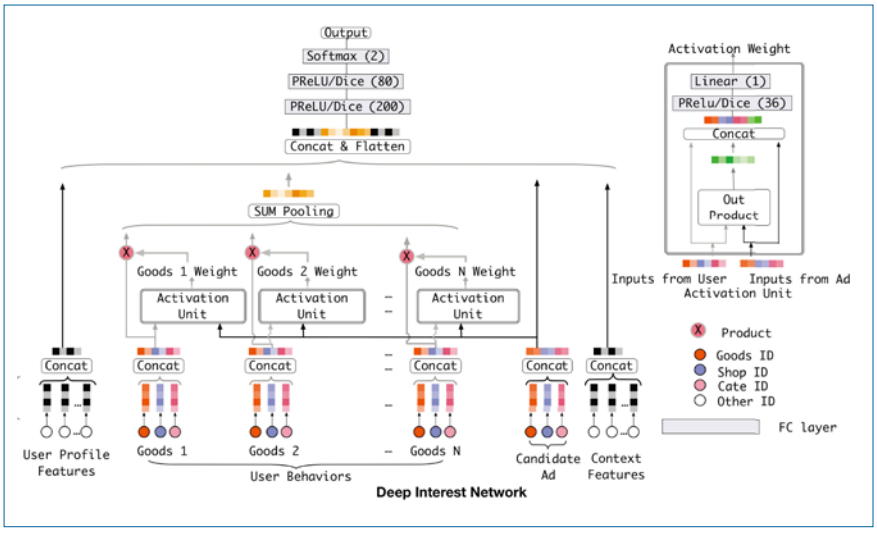

图 4　阿里巴巴广告深度兴趣网络（DIN）模型 [3]

本文提出的 EFLOPS 集群架构，从芯片到系统，从硬件到算法，综合稠密计算和稀疏计算的特征，开展端对端全栈优化设计。测试结果表明，针对拍立淘千万分类应用（视觉类），相比现有的 GPU 集群，EFLOPS 集群的通信性能提升 3.93~6.79 倍，端对端应用性能提升 1.35~2.18 倍。自然语言处理领域的 BERT 模型在 EFLOPS 集群实现了近线性扩展，在 512M 内存卡规模下，仍能保持 86% 的线性加速率。广告模型的性能更是取得了数倍提升。

EFLOPS 集群设计思路

EFLOPS 系统在集群尺度开展计算、通信和存储的软硬件一体优化，其硬件架构优势均通过系统软件得到了充分发挥，其中通信优化偏重于硬件适配软件，计算和存储的优化偏重于软件适配硬件。

通信优化驱动了整体集群架构的设计，也正是有了极致通信的能力，使得我们得以开展计算和存储部分的优化。

1. **最优化通信**：通过软硬件协同通信系统设计，消除时间占比达 50% 的通信性能瓶颈。

2. **最优化计算**：应用软件框架定制适配集群架构，实现计算算子在 CPU 和加速器间的最佳映射匹配。

3. **最优化存储**：实现数据与计算的亲和性部署。

4. **整体拥有成本（TCO）优化**：在数据中心 TCO 模型（Capex + Opex）中，当前加速器占据了 AI 集群相当比例的购置成本。使用同型号加速器条件下，构建集群的 Capex 差异不大，TCO 降低的关键在于 Opex 成本。我们一方面以异构计算资源池技术提升资源利用率，实现开源；另一方面以自研浸没式液冷技术大幅降低非计算电力消耗（在 PUE（Power Usage Effectiveness，电源使用效率）要求不高的场景仍采用风冷），实现节流。

极致通信：算法与架构的协同

EFLOPS 从降低静态延迟、降低动态延迟、无阻塞通信算法和缩减通信量四个方面，全面提升通信的效率。

1. **高带宽、低延迟的网络**。阿里巴巴通过自研流量控制、智能连接切换等核心技术，实现了具有 100 Gb/s 微秒级通信能力的全球最大规模成熟商用 RDMA 网络。EFLOPS 集群的实践证明，RDMA 以太网可以很好地满足人工智能类应用需求，且具有开放生态、易运维、高性价比优势。

2. **网络化服务器架构设计**。为避免拥塞引入的动态延迟，EFLOPS 采用加速器与网络直通的服务器架构来解决服务器内拥塞延迟，即每个加速器都自己独占高性能网卡，避免传统服务器中加速器对网卡带宽的争用，也规避了服务器内部 PCIe（Peripheral Component Interconnect express，高速串行计算机扩展总线标准）无服务质量（QoS）能力、跨根复合体（root complex）性能下降等缺陷。

3. **协同硬件架构的无拥塞通信算法**。当前的 AI 训练仍然以数据并行为主，而数据并行模型的关键通信模型就是 AllReduce，基于环形（ring）的 AllReduce 被广泛采用。但是环形算法的通信步数

与参与通信的服务器数成正比，若参与的服务器数为 N，则通信步数为 $4N$。而近期新提出的 Halving-Doubling AllReduce 算法仅需要 $2\lg(N)$，且通信传输的数据量与环形算法相同，匹配 EFLOPS 集群大规模可扩展的设计目标。由此本文提出 Rank Remapping 算法，结合 EFLOPS 的 BiGraph 网络拓扑，实现了 Halving-Doubling AllReduce 与物理拓扑的完美映射，达到与全互连拓扑相近的 AllReduce 性能（所用线缆数仅为全互连的约 1/4）。以类似的思路，我们还实现了其他常用的集合通信原语。

4. 算法优化缩减通信量。传统的并行算法会每次迭代同步梯度，而我们会在本地工作服务器上运算若干步后把模型再同步，因此相比传统算法，网络传输次数降低数倍，我们用数学证明了它对于许多模型的收敛性。但鉴于它是一种精度有损的方案，更多应用在对速度有极高要求的场景。与该工作相关的研究论文[4]已被 AI 顶级会议 AAAI 2019 录用。

计算流水线与架构的协同

AI 计算框架介于模型算法和集群架构之间，向上为模型的计算提供自动化的支持，向下对集群的资源进行管理和调度。然而，AI 计算框架本身对模型特征和集群架构并不感知，通用型的计算图处理流程往往会导致系统通用计算和加速计算资源利用率的不平衡，例如，计算框架通常采用 CPU 来进行稀疏特征的处理，但这使得系统的效率受限于 CPU 算子，也会浪费异构计算资源。为了使计算流水线的运行效率最大化，我们做了细粒度的流水线与架构协同，根据各级算子计算特征，以及通用计算和异构计算的资源利用率，分别开展 CPU 或加速器定制实现。由定制算子重新构成的流水线兼顾了速度和资源利用率。

最优化存储

提升数据访问局部性是存储优化的通用原则，AI 计算集群的存储层次较传统集群更为复杂，包括加速器内部存储（比如 GPU 显存）、服务器内存、服务器本地存储、分布式持久化存储等多个层次。

在集群层次，依托 EFLOPS 系统的通信优化，我们得以解决分布式缓存节点间的通信瓶颈，进而将数据根据访问亲和性，缓存放置到对应的存储层次中。

在服务器层次，默认采用亲和性原则，如被 GPU 处理的数据采用 GPUDirect RDMA 技术直接存入到 GPU 显存中。但加速器的显存容量往往相对较小，无法缓存全部的模型数据和临时数据。此时，需要根据模型数据的特征，对数据处理进行细粒度分解，例如我们将元数据和数据进行分拆，通过异构部件加速元数据的处理，缓减系统瓶颈。

TCO 优化

异构计算集群的加速器利用率整体偏低是行业一大痛点。EFLOPS 集群在阿里云的云原生技术体系下，自主研发了底层资源池技术（如 GPU stream、GPU 细粒度切分）来实现资源超卖，从而提升整体资源利用率。我们自研的加速硬件虚拟化技术，实现了 GPU 显存和算力双维度的自由切割，做到资源强隔离，用户零感知，且无额外性能开销，其中的 Activation on Allocation（AA）模式从框架层面延迟 GPU 资源绑定，从而达到 GPU 超卖的效果，双十一期间 GPU 资源利用率提升了数倍。

此外，EFLOPS 系统中 GPU 绑定网卡和扁平拓扑的架构，模糊了服务器内和服务器间通信的边界，可以更好地发挥 GPU 池化的效果。

AI 计算集群所引入的异构加速部件（比如

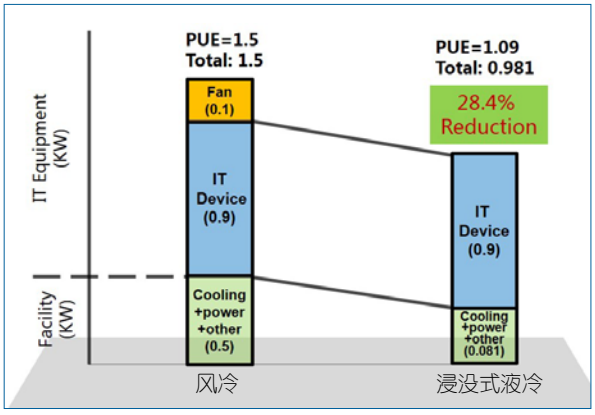

图 5　浸没式液冷与风冷 PUE 对比

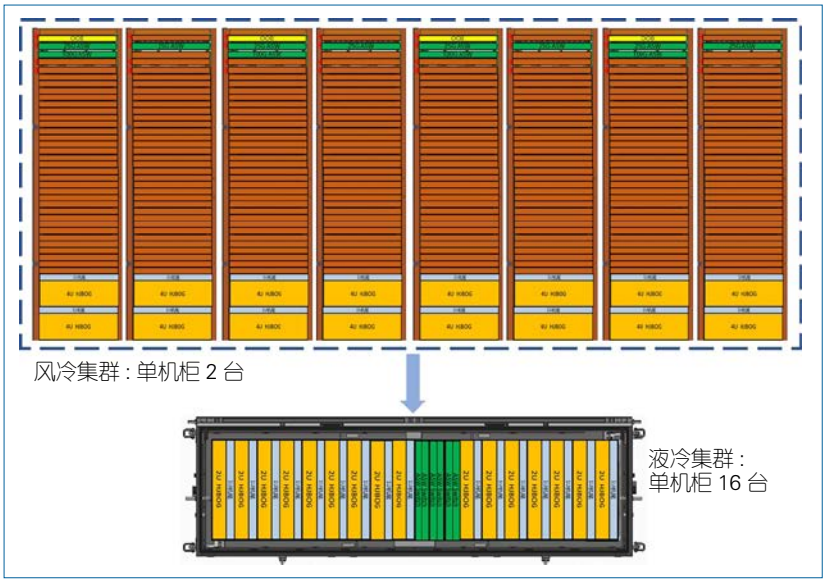

图 6 浸没式液冷的 IDC 空间节省

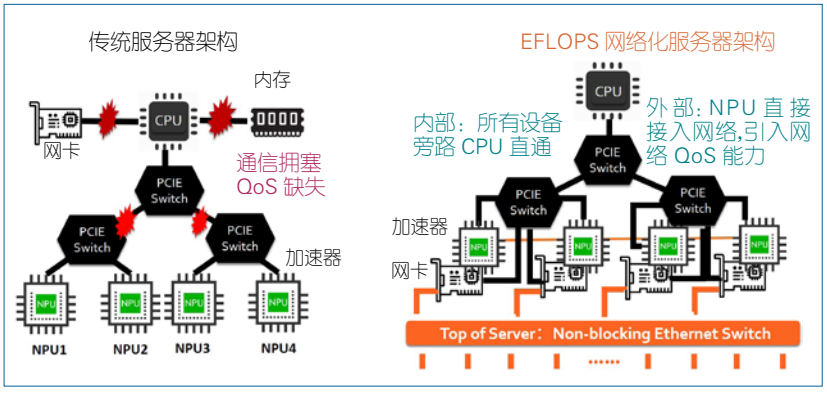

图 7 EFLOPS 服务器架构图

GPU）导致了系统功耗的显著增加，功耗成本不容忽视。高功耗不仅意味着会耗费更多的电能，也会给电源布线、机房通风、空调散热等各方面带来巨大的压力，导致系统部署和维护成本的增加。EFLOPS AI 集群采用阿里巴巴自研浸没液冷技术对现有 GPU 服务器和数据中心架构进行重构及融合创新，将 PUE 年均值降至 1.09，逼近理论极限值 1.0。也就是说，当有一度电用于计算时，耗费在 GPU 服务器散热上的电只有 0.09 度，大大提升了数据中心的能源使用效率。对比当前风冷技术，整体能耗节省比例近 30%，详细对比如图 5 所示。

阿里巴巴的浸没式液冷技术将单机柜功率密度从 15 kW 提升到 100 kW 以上。如图 6 所示，相当于 8 个风冷机柜的服务器（单机 8 块 GPU 卡）合并到一个液冷机柜，进一步降低了互联网数据中心（IDC）空间资源的 CapEx（Capital Expenditure，资本性支出）投入。

EFLOPS 集群架构

EFLOPS 服务器架构

如图 7（左）所示，传统服务器架构的瓶颈主要来自内部 PCIe Fabric 树形互连。

首先，传统的数据中心服务器通常只配备一个网络接口（独立网卡或者 Bond 网卡），当该服务器配备多个加速部件（比如 GPU）并通过网络接口并发传输数据时，就会面临很大的流量汇聚，使其成为系统的瓶颈。而这种同步式的网络访问，在分布式 AI 训练任务中非常常见。AI 训练的数据集一般被划分为多个批次，每个批次的数据处理完成之后，所有参与计算的加速器（本节的加速器以 NPU 标识）都要进行梯度的同步。跨服务器的 NPU 梯度同步操作都要通过网络接口进行通信。这种周期性的同步式网络接口访问，势必导致网络接口上的拥塞。

类似的端口拥塞还会发生在 PCIe 树形拓扑的跟节点处。分布式 AI 训练业务在每个批次的数据处理完成之后，会同步载入下一批次数据，导致内存的并发访问。

其次，PCIe Switch 端口上的拥塞可能导致整体通信效率的降低。如图 7（左）所示，当 NPU1 和 NPU3 同时向 NPU2 发送数据时，将会在与 NPU2 直接相连的 PCIe Switch 端口上形成拥塞。由于

NPU1 和 NPU3 到 NPU2 的通信距离不同，导致二者之间具有显著的带宽差异。而 AI 训练任务的梯度 AllReduce 是一个全局性的同步操作，其完成时间往往受限于最慢的链路，所以这种链路带宽的不公平性也会导致系统性能的下降。

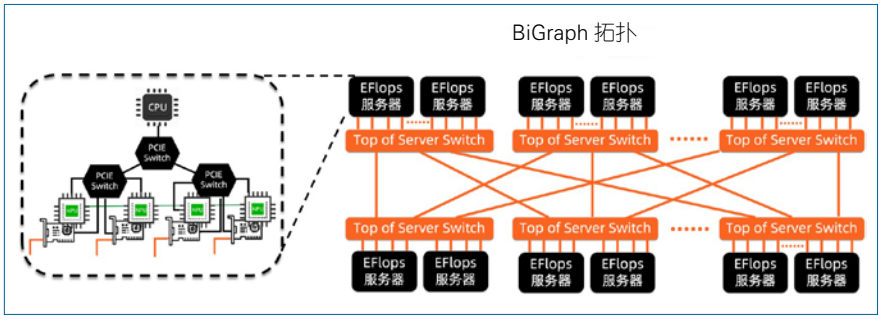

图 8　EFLOPS 系统互连架构

最后，出于种种原因，PCIe 交换芯片往往只会实现一个虚拟通道，导致 QoS 能力缺失，这就使得服务器内各种流量没有隔离能力，形成带宽的无序争抢。

EFLOPS 服务器架构重点解决上述互连问题，如图 7（右）所示，其配备了与加速器（NPU）等量的网卡（NIC），并将 NPU 和 NIC 进行绑定配对，每一对绑定的 NPU 和 NIC 处于同一 PCIe Switch 之下，约束 NPU 的网络通信只能经由自己绑定的 NIC。这样，NPU 的网络通信流量全部被局限在 PCIe Switch 之内，避免了网络接口上的拥塞。针对 PCIe Switch 引入的拥塞问题，在 PCIe 流量较大的情况下，禁用 NPU 之间进行跨 PCIe Switch 通信，使其通过网络接口进行数据交换，利用网络协议栈的流量控制机制来降低系统的拥塞程度。

值得强调的是，本文的网络化服务器架构是一个开放的架构，可为各种加速器提供高速互连，对于自带直连总线（如英伟达的 NVLink）的加速器同样兼容，利用其直连总线实现更高带宽通信。

系统互连架构

数据中心大多采用 Clos[1] 拓扑，提供了高对剖带宽、可扩展的基础通信能力，但由于路径选择的哈希算法总是存在碰撞的可能，使得网络中的拥塞无法避免。相比传统仅优化拥塞控制算法的思路，EFLOPS 从更上层架构进行网络流量管理，以彻底解决网络的拥塞问题。

如图 8 所示，配合 EFLOPS 多网卡服务器结构，本文提出了 BiGraph 扁平化拓扑，分为上下两组，每组的交换机与另一组交换机全互连，同组交换机之间的数据交换需要另一组交换机转发，这样每一个交换机都扮演了 Clos 网络中的 Spine（脊柱）和 Leaf（叶子）两个角色，最大跳步数仅为 3，BiGraph 拓扑具有如下两个重要的特性。

1. 它在两层交换机之间提供了丰富的物理链路资源。在 N 个计算服务器的系统中，两层交换机之间至少存在着 $N/2$ 个物理链路可供使用。这意味着我们有机会将 Halving-Doubling AllReduce 算法 [5] 的所有连接一一映射到可用的物理链路上，避免它们之间的链路争用，以彻底解决网络拥塞问题。

2. 接入不同层次的任意两个计算服务器之间的最短路径具有唯一性。我们可以充分利用这一特性，在通信库甚至更高层次进行服务器间通信模式的管理。比如，在建立连接的时候，选择合适源和目的服务器，来控制网络上的路径选择。

通信库软件是发挥 BiGraph 拓扑优势的关键，阿里巴巴自研了 ACCL（Alibaba Collective Communication Library）集合通信库，首先它在物理网络中构建出 BiGraph 虚拟拓扑，然后基于该虚拟结构，实现我们提出的无拥塞集合通信算法。无拥塞集合

[1] Clos 是一种多级交换架构，目的是为了在输入输出增长的情况下尽可能减少中间的交叉点数。1953 年，贝尔实验室查尔斯·克洛斯（Charles Clos）博士在《无阻塞交换网络研究》论文中提出了这种架构，后被广泛应用于 TDM 网络（多半是程控交换机）。为纪念这一重大成果，便以他的名字 Clos 命名这一架构。

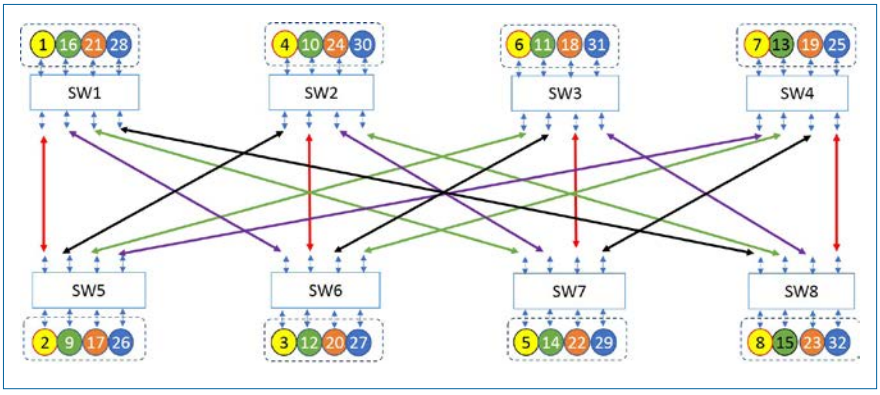

图 9　Re-ranking Halving-Doubling 算法示意图

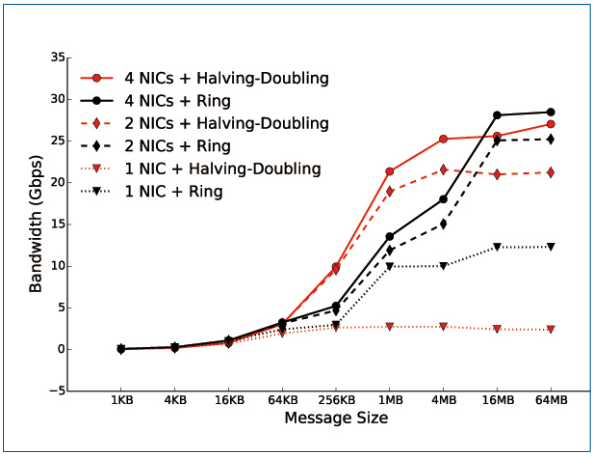

图 10　通信性能对比

通信算法是我们在标准 Halving-Doubling 算法的基础上，提出的一套新的 Re-ranking Halving-Doubling 算法，实现了通信连接与 BiGraph 拓扑的完美映射，从根本上避免选路冲突。

相比最流行的 Ring AllReduce 算法，EFLOPS 的算法更利于大规模性能扩展。Ring AllReduce 非常适合传统单网卡服务器架构，每一步需要传输的数据量少而且采用单向环式的网络传输，但需要 $O(N)$ 步执行，延迟随系统规模扩大而线性增加。Halving-Doubling 算法则是通过递增和二分的方式快速地实现数据传输，仅需要 $O(\log N)$ 步，但每一步要传输的数据量比 Ring AllReduce 更大，这一特征恰好与 EFLOPS 的互连网络能力适配。

我们提出的 Re-ranking Halving-Doubling 算法的核心是根据每个进程的物理位置，重新排列该进程对应的排名，结合节点之间的同步策略，使得任何时刻任何点到点的数据传输都能独占一条物理链路，从而有效地避免了网络拥塞，理论上能够达到线速的传输。

图 9 以 8 台服务器，每台服务器包含 4 个加速器的系统为例，对该算法进行说明，其中方形表示交换机，圆圈表示加速器，圆圈里的数字表示重新排列后新的排名。图中的连线代表交换机之间的物理连接，不同颜色代表不同步骤下使用的路径。按照图示重新排列后，可以看到算法的任何一个步骤，同一个主机的四个加速器走的都是不同的直连链路，这样保证了数据经过的路径最短，且加速器间的数据传输路径没有冲突。

EFLOPS AI 集群性能

只要在多机多卡环境，不需要太大的规模，EFLOPS 集群架构就可以发挥明显的性能优势。在一个 64 张 GPU 卡（NVDIA V100 32G 显存）的小规模集群中，我们开展的 AllReduce 集合通信性能测试表明，采用不同的 AllReduce 算法，EFLOPS 集群的硬件设计可以将通信效率提升 2.3~11.3 倍。EFLOPS 算法架构协同算法可以将通信效率进一步提升 1.4~7.3 倍。随着系统规模的增长，网络拥塞概率的增加，EFLOPS AI 集群的通信性能优势更明显，如图 10 所示。

在该 64 GPU 卡测试场景中，拍立淘百万分类大模型的端到端性能提升了 2.2 倍。对自然语言处理领域广泛应用的 BERT 预训练模型进行评测的结果如图 11 所示，在 EFLOPS 集群中，BERT 的通信开销得到了大幅降低，仅使用 EFLOPS 硬件即可获得 2 倍通信性能的提升，叠加 ACCL 通信库支持，整体性能提升了 2 倍，通信性能提升了 4 倍。

我们对典型的稀疏计算模型——点击率预估模型进行了评估。该类模型同时存在多机多卡的异

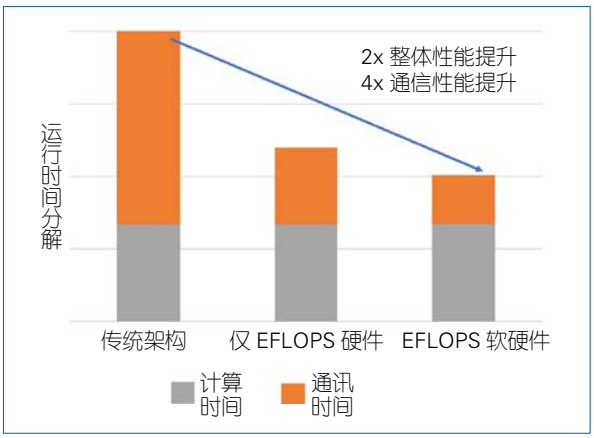

图 11　BERT 性能评价

步训练和同步训练需求。针对异步训练，主要使用 PS 构架，一个 PS 服务有几十个工作服务器。受到模型精度影响，每个工作服务器只能使用小的批量样本数量（batch size），计算加速比较低，因此 CPU 与加速器的配比需求更均衡。针对该场景，EFLOPS 集群大幅降低了单服务器加速器数量，以达到最佳性价比，在实际生产业务中已获得了数倍的训练性能提升。在同步训练中，由于每个工作服务器之间的训练保持同步，模型能更快收敛，最终精度更高。对于同步的开销，我们通过个性化定制的流水线（pipeline）架构实现了隐藏，原来在上百台机器上完成的训练，现在用 8 台 EFLOPS 服务器就能完成，同时还有性能收益。

未来方向展望

大规模 AI 应用的发展仍将是驱动智能计算集群演进的重要力量。一方面，AI 自身仍然在不断演进，GPT-3 等超大规模模型的出现，使得数据和模型混合并行成为趋势，不断推高对计算和通信性能的需求；另一方面，随着 AI 在传统领域的深度渗入，“AI+大数据 + 高性能计算”的混合计算模式将改变应用负载的计算和通信特征，智能计算集群的设计，必须深度依靠真实场景的打磨。

此外，智能计算集群将对加速芯片和网络硬件的发展起到牵引作用，成为新硬件的试炼场和孵化器。具体地说，网络与计算将进一步深度融合发展，一方面，加速器性能的提升，将不断推动网络领域的创新，除了更高带宽、更低延迟的物理网络，面对多任务、多租户应用场景的可预测网络服务性能将是刚需；另一方面，网络设备能力的提升，将推动一些整体协同性计算任务向网络设备迁移，在网计算可以逐渐看到实际应用的价值。

最后，为应用提供最优性价比的算力，集群尺度的软硬件一体化设计是必须坚持的方向，从更丰富的软硬件层次和更大的资源池中找到最优解，避免昂贵硬件的过度设计，避免算力的孤岛化。 ■

曹　政
CCF 专业会员。阿里云基础设施事业群资深技术专家。负责面向 AI、大数据、高性能计算的数据中心研发。
zhengzhi.cz@alibaba-inc.com

董建波
阿里云基础设施事业群高级技术专家。负责高性能 AI 计算集群通信系统研发。
jianbo.djb@alibaba-inc.com

金铃铃
阿里云基础设施事业群高级技术专家。负责高性能 AI 集群性能优化。
l.jin@alibaba-inc.com

其他作者：钟杨帆

参考文献

[1] 陈左宁. 人工智能进展对算力需求分析. HPC China 2020.

[2] Google. https://cloud.google.com/blog/products/ai-machine-learning/googles-scalable-supercomputers-for-mach -ine-learning-cloud-tpu-pods-are-now-publicly-available-in-beta.

[3] Zhou G, Zhu X, Song C, and et al. Deep interest network for click-through rate prediction[C]// Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018:1059-1068.

[4] Yu H, Yang S, Zhu S. Parallel Restarted SGD with Faster Convergence and Less Communica-tion: Demystifying Why Model Averaging Works for Deep Learning[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33:5693-5700.

[5] Thakur R. Optimization of Collective Communication Operations in MPICH[J]. International Journal of High Performance Computing Applications, 2005, 19(1):49-66.