## DistGnn：大规模图神经网络的可扩展分布式训练

### 1 Motivate：

#### 1.1 目标

通过高效共享内存实现CPU集群的full-batch训练，使用最小顶点切割图划分算法来减少通信，使用一系列延迟更新算法来避免通信。

#### 1.2 相关工作

现阶段的挑战是在大规模图上进行训练，有两种方式：分布式的和mini-batch。mini-batch基于邻居采样的，但论文说，在某些情况下，基于邻居采样的精读低于full-batch训练。

GPU-based分布式大规模GNN系统：NeuGraph、Roc、CAGNET（已读）

CPU-based分布式大规模GNN系统，迄今为止（2021）提出的算法都是min-batch的：AliGraph（它支持分布式存储、采样和聚合算子，以及一套图分割技术，包括基于顶点和边缘切割的技术。AliGraph实现并展示了缓存重要顶点的邻居以减少通信负载的有趣概念的有效性）、DistDGL（它保存分布式数据服务器中的顶点特征，可以查询这些特征以进行数据访问。由于其效率低下的图形采样操作的执行时间较长，它设法将通信与采样时间重叠）

#### 1.3 full-batch训练的挑战（与传统的DL相比）

* 通信量增加，需要通信顶点的特征向量和参数梯度。
* 较低的矩阵密度和训练操作的顺序性使得计算和通信重叠。
* 聚合操作往往受到内存和通信带宽的限制

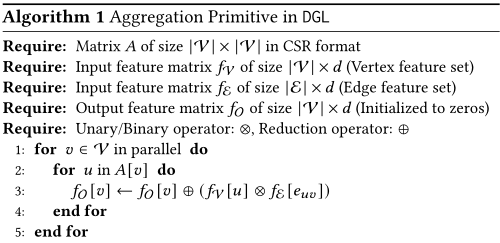
在分布式中，图分区在通信中起着至关重要的作用

#### 1.4 主要贡献

* 作为单机自定义SpMM的聚合原语的多个变体，实现了一个高度优化的架构。在不同数据集下的GrapgSAGE算法。比DGL实现各种快啊。
* 将noval基于顶点切割的图划分算法应用于大规模图，旨在减少分布式full-batch训练CPU之间的通信。
* 将noval的延迟更新算法应用在feature聚合中，来实现最佳的通信避免。将通信与计算重叠，跨epoch传播。这种重叠是以牺牲feashness为代价的（即使用部分过时的、旧的远程顶点的feature）。
* 首次在基于CPU的分布式内存系统上演示了full-batch GNN训练。

### 2 共享内存算法

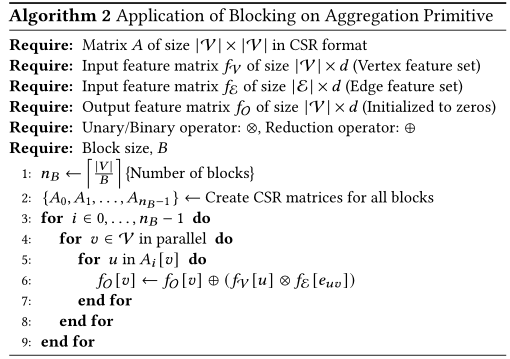
#### 2.1 DGL的聚合原语

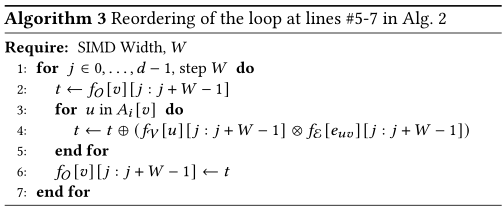


#### 2.2 高效的聚合原语（本文提出）

真实世界的图都是高度稀疏的，并且对于大规模图，顶点和边的feature矩阵不适用于cache。为了聚合顶点的特征向量，必须访问其所有邻居和相应边的feature向量。顶点入边的feature向量连续存储，但在Alg.1中只访问一次，可以在存储带宽的限制下一次访问足够大的内存块；另一方面，顶点v邻居可以是非连续的、稀疏的，其使用的次数也是多次的。

因此，可以应用缓存block来利用上述的缓存重用。Block的大小选取是一个挑战。



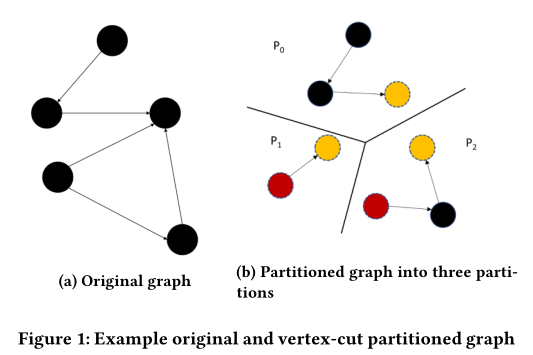


### 3 分布式内存算法（数据并行）

#### 3.1 图分区

对图分区以实现最小化通信。使用顶点切割的方式划分图数据，每个边只存在于一个分区之中，而一个顶点可以存在于多个分区之间，但是对每个点的更新都必须将其他分区的这个点进行克隆，这种克隆的数量称为复制因子。分区的目标：减少分区之间的通信（通过降低复制因子来实现），生成平衡分区（使用均匀边分布来作为负载平衡的度量）。

分区包含两种类型的点——分割顶点和非分割顶点。每个顶点有两个ID——全局ID和局部ID，全局ID存储每个分区局部ID的范围。



#### 3.2 Delayed Remote Partial Aggregates（三种算法）

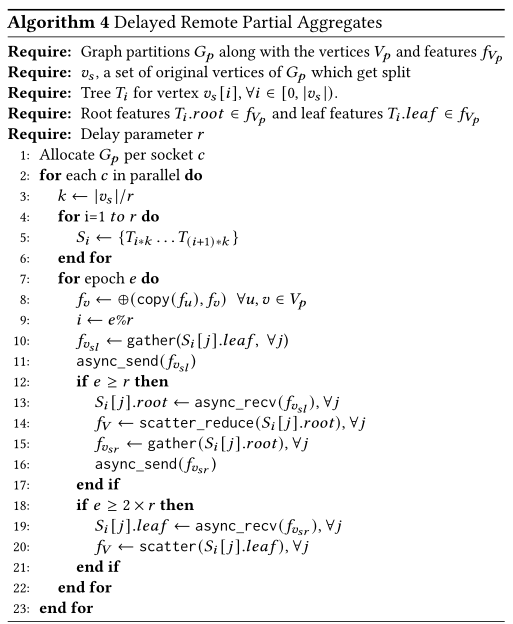
0c：每个分区、每个layer，忽略分割节点的通信，只在本分区内部做聚合，论文说这个方法最快，而且准确率最乐观（暂时不太懂）。

cd-0：每个分区、每个layer，在局部聚合之后，将局部聚合信息通信，所以每个分割节点也可以聚集到所有邻居的feature。这种方法的计算准确率与单GPU相同。

cd-r：通信避免机制，将远程聚合通信与本地计算重叠，在一个epoch的连续步骤之间的依赖没有epoch内的重叠，这是因为采取了跨eopch（inter-epoch）计算通信重叠的方式。每个分割节点在epoch i开始通信，在epoch i+r时异步接收并处理聚合

Alg.4:r = 0:cd-0; r>0:cd-r 忽略通信:0c

（看完算法感觉r=1是cd-0吧，r>1是cd-r，有点懵逼）



随机一个分割节点为树根。