1. 单机多卡

单机多卡通常采用的是数据并行的方式进行分布式训练。在这种模式中，不同的GPU有同一个模型的多个副本，每个GPU分配到不同的数据，然后所有GPU的计算结果按照某种方式合并。

图片包含 屏幕截图

描述已自动生成

数据并行这种分布式训练模式有如下步骤：

1. 将数据分成多份，分别传输到各个进程
2. 各进程开始训练（计算梯度）
3. 训练完后，将结果同步到其他进程，待同步完成后，每个进程拥有所有的结果
4. 各个进程合并结果（梯度平均）并更新模型
5. 重复以上步骤

数据并行在TensorFlow对应的策略是MirroredStrategy。

1. Parameter Server

在TensorFlow中，提供了另一种能在多机多卡上训练的策略ParameterServerStrategy。在这种策略中，一些进程称为worker，专门用于梯度计算；另一些称为parameter server(ps)，负责计算平均梯度并更新模型。

图片包含 文字

描述已自动生成

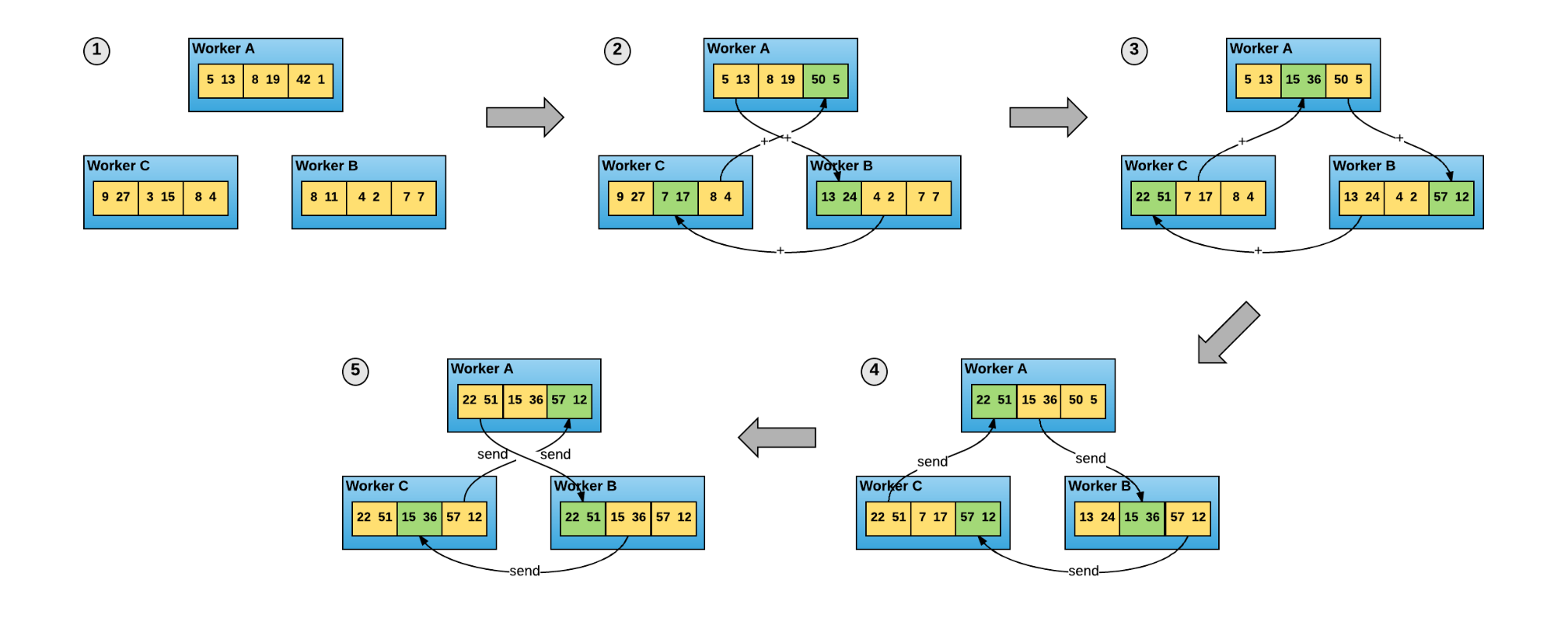
如上图所示，所有的worker计算完成后，将结果同步到ps，由ps来计算平均梯度并更新模型，之后，各个worker拉取最新模型参数，进行下一轮计算。对于这种策略，无论是只有一个还是有多个ps，在大规模的分布式训练中，ps与worker间的网络通信将会是性能瓶颈。

1. Ring AllReduce

基于以上原因，Uber研发了新的集群方案：Horovod。它采用了全新的梯度同步算法，叫ring-allreduce。此算法各个节点只与相邻的两个节点通信，并不需要参数服务器。

一次权重更新，主要包含两个过程：

1. 累计梯度
2. 梯度分发到所有节点



如上图第2、3步所示，将所有梯度分成n个片段，每次只与相邻节点传递1个片段的梯度，第n-1次后，每一片段的梯度都完成了累计，但不同的片段累计值分布在不同节点上。

将上一步累计的梯度再通过n-1次的相互交换后，所有节点的梯度完成同步，如上图4、5步所示。梯度同步完成后，再各自平均，更新参数，这样所有节点权重就更新完成。

1. 参考
2. <https://www.infoq.cn/article/J4ry_9bsfbcNkv6dfuqC>
3. <https://eng.uber.com/horovod/>
4. <http://sharkdtu.com/posts/dist-tf-evolution.html>
5. <https://fyubang.com/2019/07/08/distributed-training/>
6. <https://www.tensorflow.org/guide/distributed_training>