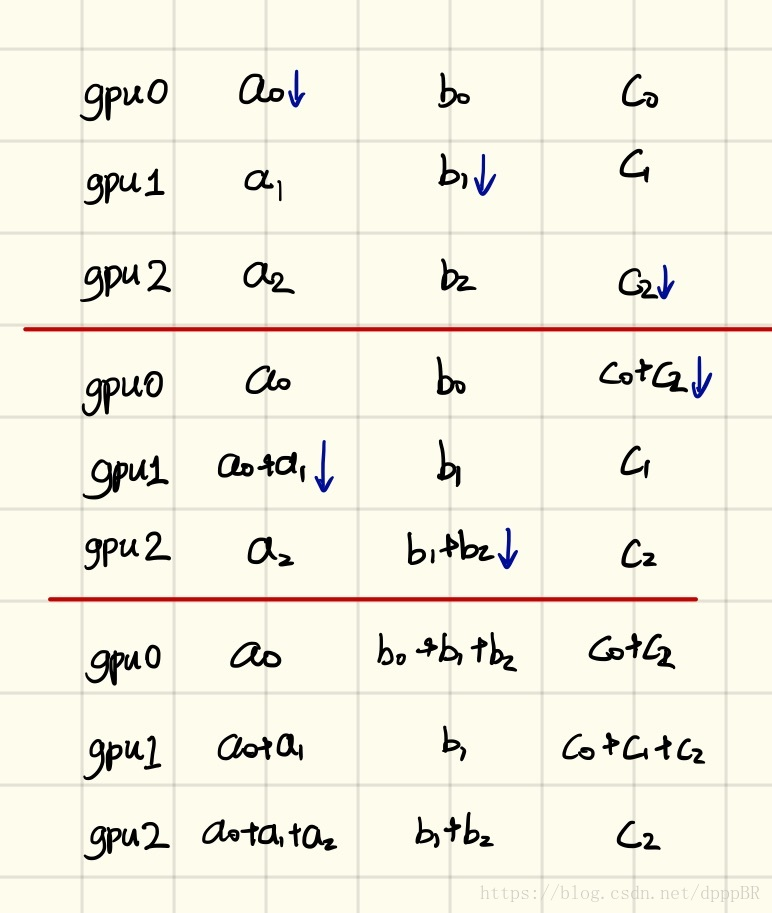
# ring-allreduce图解

ring allreduce和tree allreduce的具体区别是什么？ - 知乎  
<https://www.zhihu.com/question/57799212/answer/292494636>  
ring-allreduce简介 - Brassica\_的菜园 - CSDN博客  
<https://blog.csdn.net/dpppBR/article/details/80445569>

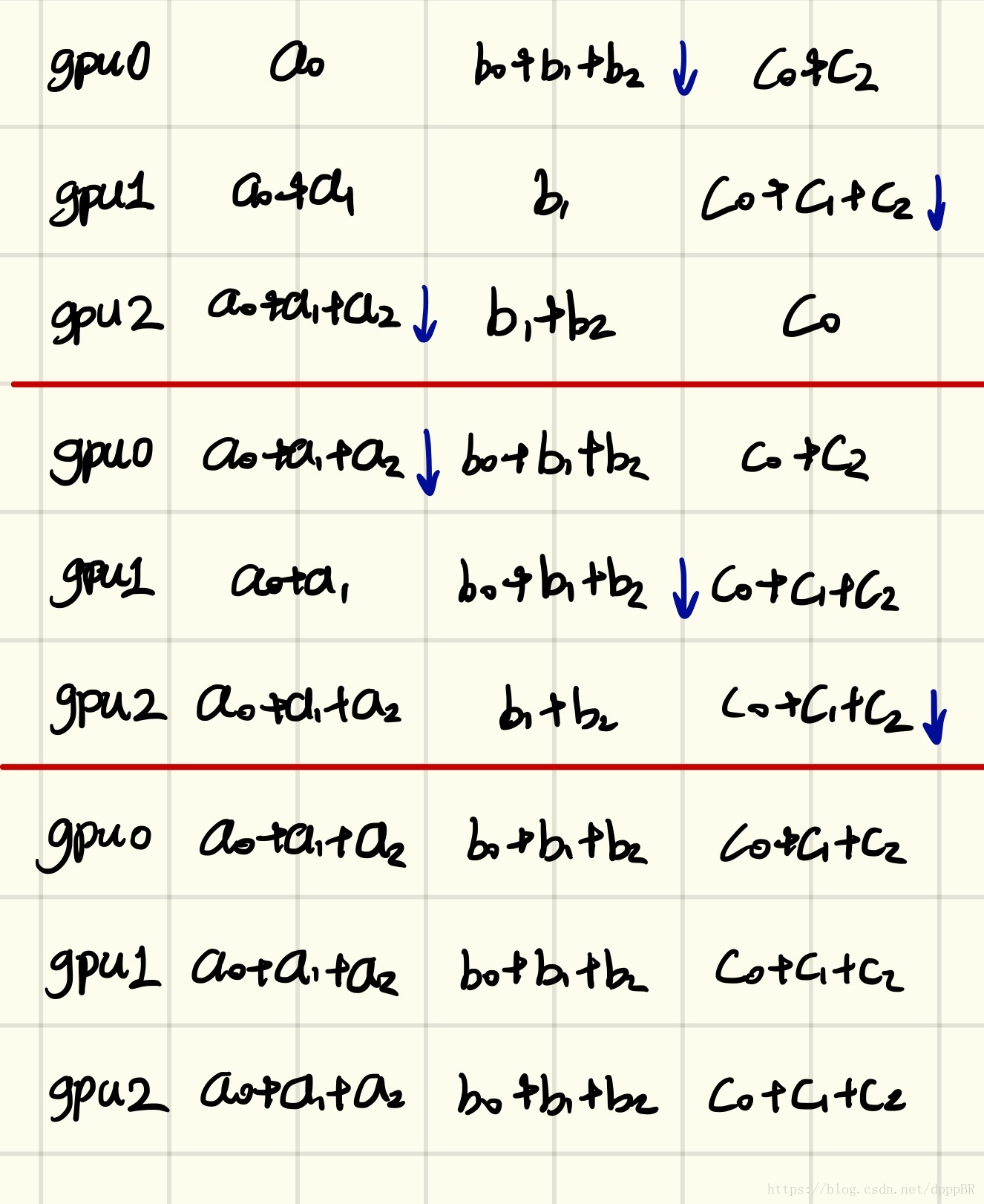
多卡gpu训练的缺陷：每次需一个gpu从其他gpu上收集训练的梯度，然后将新模型参数分发到其他gpu。最大的缺陷是gpu 0的通信时间是随着gpu卡数的增长而线性增长的。

所以就有了ring-allreduce。该算法思想：取消Reducer，让数据在gpu形成的环内流动，整个ring-allreduce的过程分为两步：第一步是scatter-reduce，第二步是allgather。

举一个3gpu的例子。首先是第一步，scatter-reduce：

[](https://user-images.githubusercontent.com/7320657/53217738-3db76680-3694-11e9-8076-ce454f544b24.png)

**第二步，allgather：**

[](https://user-images.githubusercontent.com/7320657/53217751-49a32880-3694-11e9-84ec-0e104f9173b7.png)

# 二、Ring-Allreduce

Ring-Allreduce是一种以环状拓扑为基础的通信系统。整个体系结构的工作过程见下图，Rank代表了各个 GPU的进程编号，并且梯度信息可以在两个不同的区域中同步传输。在Ring-Allreduce体系结构中，每台计算机都是一个工作节点，按环形排列。

Ring-Allreduce体系结构的工作过程被分成两个阶段，即Scatter-Reduce和 Allgather。在Scatter-Reduce阶段，完成了数据的分配与并行，各个工作节点之间的数据交换。最后，在每一个节点上都会有一个最终的结果。Allgather阶段实现了数据的整体同步和压缩，每一个工作节点之间都会进行一些最后的处理，这样对于所有节点来说就可以得到一个完整的结果。

## 1. Scatter-Reduce阶段

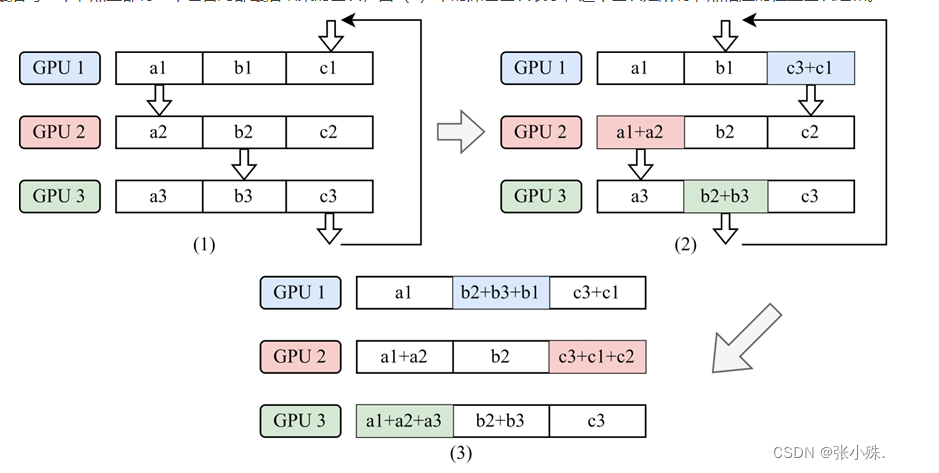
Scatter-Reduce阶段：假定这个阶段的目的是求和，在这个系统中有 N个工作结点，每一个结点中的数据量大小都是K，在Scatter-Reduce的后期，每一个结点都有一个包括初始数组和的而且大小相同的矩阵。

具体的，

第一步，每个结点把本设备上的数据分成 N个区块， N是Ring-Allreduce体系结构中的工作节点数目，见下图步骤（1）。

在第二步，在第一次传输和接收结束之后，在每一个结点上累加了其他节点一个块的数据。这样的数据传输模式直到“Scatter-Reduce”阶段结束，见下图步骤（2）。

最后每一个节点上都有一个包含局部最后结果的区块，由（3）中的深色区块表示，这个区块是所有节点相应的位置区块之和。



## 2. Allgather阶段

Allgather阶段：每个工作节点将包含最终结果的块数据块交换, 这样所有的结点就会得到一个完整的结果，

Allgather阶段总共包含有数据发送和接收N一1次，不同的是，Allgather阶段并不需要将接收到的值进行累加，而是直接使用接收到的块内数值去替环原来块中的数值。在迭代完第1次这个过程后，每个节点的最终结果的块变为2个，如图3.3步骤（2）所示。

之后会继续这个迭代过程直到结束，使得每一个节点都包含了全部块数据结果。下图为整个Allgather过程，可以从图中看到所有数据传输过程和中间结果值。

