大模型分布式训练故障恢复篇

来自： AiGC面试宝典

宁静致远

2023年12月24日 00:35

一、为什么 大模型分布式训练 需要 故障恢复?

大规模分布式训练场景由于集群规模过大，芯片设备、主机、网络等均会不定期出现故障，如果此时想要继续训练，那么就需要从上次存储的ckpt进行恢复，然后resume training。这个过程中产生的时间间隔就是集群故障带来的开销，虽然不可避免，但是我们可以尽可能的减少故障带来的影响。

二、如何获取最优的ckpt存储间隔?

假设我们均匀的同步存储ckpt，那么这个时候我们就需要根据集群环境获取最优的ckpt interval，首先集群时间损失可以做如下定义，故障随机发生在ckpt interval区间：

```
集群时间损失 = ckpt存储耗时 + 故障期望次数 * 恢复训练耗时（ckpt interval/2+恢复训练耗时）
```

通过导数为0，可以根据集群环境，得到对应最优的ckpt interval，当然ckpt interval肯定是远大于1的。

三、ckpt存储能否实现异步或者部分掩盖?

异步存储ckpt的最大问题：**设备内存踩踏**，如果在另外一个stream里做D2H数据拷贝，同时模型训练过程继续运行，那么就会有一个非常尴尬的问题，就是所有参数的D2H操作还没有完成，这时候下一个step已经开始更新参数或优化器状态，那么后面没完成的操作就会拷贝错误的数据。
由于ckpt存储时间不可控，不能确定是否小于下一个step的执行时间，所以内存踩踏的问题不可避免，即完全异步的方案是不可行的。如果要想做到部分掩盖，**本人认为可以有如下两个方案供选择**：

- 在训练脚本侧修改，在下一次更新参数或优化器状态之前，强制等待ckpt存储完成，这样可以尽可能的overlap；
- 随便yy一下，在框架侧修改，比如H2D non-blocking操作在后续有数据依赖的时候，会强制加sync point，框架侧也可以新增一个D2H拷贝，在后续有数据写操作的时候，强制添加sync point。

四、断点续训/临终遗言是否真实可行?

绝对可行，但有一点受限。大模型训练场景多是DP/TP/PP多维并行场景，任意一个节点出现故障的可能性都是存在的。如果任何一个PP stage都存在一个完整的TP Group，就是该rank对应的节点没发生故障，那么整网参数就是完整的，可以在框架侧捕获分布式error做临终参数存储，这样ckpt interval就趋近于0。如果不满足整网参数完整这个条件，那是做不到临终存储整网参数和优化器状态。根据经验，框架侧开发并不会很难，需要结合rank编排做定制研发。当然如果故障发生在参数或存储器状态更新的时候，那也是无法保证整网参数完整性的，这种情况也不能做临终处理。

对于临终遗言/断点续训，基于训练框架对深度学习框架做深度定制是比较好的出路。