在horovod下使用多机多卡需要满足以下3个先决条件：

1. 不同机器可以访问相同的文件：nfs
2. 不同机器使用相同的训练环境: Docker
3. 不同机器可以ssh交互：ssh 免密登录

假设现在要在两台服务器A和B上多机多卡跑horovod，A为主worker，下面介绍怎么准备horovod的启动条件。

**NFS**

**在A上的操作**

# 在A上的操作

#1. 安装nfs服务器

sudo apt install nfs-kernel-server

#2. 编写配置文件

sudo vi /etc/exports

#/etc/exports文件的内容如下

/data1/share \*(rw,sync,no\_subtree\_check,no\_root\_squash)

#3. 创建共享目录

sudo mkdir -p /data1/share

#4. 重启nfs服务

sudo service nfs-kernel-server restart

#5. 常用命令工具：

#在安装NFS服务器时，已包含常用的命令行工具，无需额外安装。

#显示已经mount到本机nfs目录的客户端机器。

sudo showmount -e localhost

#将配置文件中的目录全部重新export一次！无需重启服务。

sudo exportfs -rv

#查看NFS的运行状态

sudo nfsstat

#查看rpc执行信息，可以用于检测rpc运行情况

sudo rpcinfo

#查看网络端口，NFS默认是使用111端口。

sudo netstat -tu -4

**在B上的操作**

# 在B上的操作

#1. 安装nfs客户端

sudo apt install nfs-common

#2. 查看NFS服务器上的共享目录

sudo showmount -e A的ip

#3. 创建本地挂载目录

sudo mkdir -p /data1/share

#4. 挂载共享目录

sudo mount -t nfs A的ip:/data1/share /data1/share

**Docker**

分别在A和B服务器上通过docker建立容器，这里仅以在A上的操作为例，B的类似。

1. 将用户加到docker的组

sudo gpasswd -a user docker

1. pull docker  
   docker pull horovod/horovod:0.18.2-tf2.0.0-torch1.3.0-mxnet1.5.0-py3.6-gpu
2. 建立容器  
   这里需要注意要把上一步的共享目录映射进来

nvidia-docker run -itd --net=host -v /data1:/data1 --name horovod --shm-size=126g horovod/horovod:0.18.2-tf2.0.0-torch1.3.0-mxnet1.5.0-py3.6-gpu bash

1. 运行容器

docker exec -it horovod bash

**构建训练镜像**

完成各个主机的docker 容器配置后，可以将修改后的容器保存为镜像，方便后续使用，以主机A为例，B类似。

#1. 查看运行容器，获取container\_id

docker container ls

#2. 通过容器创建镜像

docker commit container\_id 新镜像名称/TAG

#3. 查看创建的新镜像

docker image ls

镜像的导出导入，以B主机为例。

# 1. 在B操作，导出镜像打包文件

docker save lab/son | gzip > lab\_son-latest.tar.gz

# 2.scp至想要部署的主机，执行导入镜像操作

docker load -i lab\_son-latest.tar.gz

创建训练镜像后，可以通过直接创建容器开启训练

#1. 通过新镜像启动容器，注意挂载共享盘

nvidia-docker run -itd --net=host -v /data1:/data1 --name horovod --shm-size=126g 新镜像名称/TAG bash

#2. 进入镜像

docker exec -it horovod bash

#3. 在B主机操作，开启sshd

/usr/sbin/sshd

#4. 在A主机，启动训练

horovodrun -np 2 -H localhost:1,192.168.1.162:1 --start-timeout 300 -p 12345 python pytorch\_mnist.py

time horovodrun -np 3 --timeline-filename /djm/data/share --cache-capacity 0 -H localhost:1,192.168.1.109:1,192.168.1.115:1 --start-timeout 300 -p 12345 python train\_distributed.py

or

mpirun -np 3 -H localhost:1,192.168.1.109:1,192.168.1.115:1 --allow-run-as-root -bind-to none -map-by slot -mca plm\_rsh\_args "-p 12345" -x NCCL\_DEBUG=INFO -x LD\_LIBRARY\_PATH -x PATH -mca pml ob1 -mca btl ^openib -mca btl\_tcp\_if\_exclude lo,docker0 python train\_distributed.py

**补充**

* 安装Docker和nvidia-docker2可参考docker官方或horovod教程：

[https://github.com/horovod/tutorials/blob/master/fashion\_mnist/INSTALL.md](https://link.zhihu.com/?target=https%3A//github.com/horovod/tutorials/blob/master/fashion_mnist/INSTALL.md)

* 目前启动horovod需要各个主机的网卡名称相同，ubuntu下更改网卡名称参考：

[https://www.jianshu.com/p/8a469e2eab28](https://link.zhihu.com/?target=https%3A//www.jianshu.com/p/8a469e2eab28)

**注意修改/etc/network/interfaces文件时原始内容最好保留**