**Script for train model:**

**ML FLOW Tracking:**

1. **Bring Up Sever**

from chi import server, context, lease

import os, time

context.version = "1.0"

context.choose\_project()

context.choose\_site(default="CHI@TACC")

l = lease.get\_lease(f"mi\_100\_zh3194") #改成租用的名字

l.show()

username = os.getenv('USER') # all exp resources will have this prefix

s = server.Server(

f"node-mltrain-{username}",

reservation\_id=l.node\_reservations[0]["id"],

image\_name="CC-Ubuntu24.04-hwe"

)

s.submit(idempotent=True)

s.associate\_floating\_ip()

s.refresh()

s.check\_connectivity()

s.refresh()

s.show(type="widget")

s.execute("git clone --recurse-submodules [https://github.com/Jasonzzzz28/Transformer-QA")](https://github.com/Jasonzzzz28/Transformer-QA\"))

s.execute("curl -sSL https://get.docker.com/ | sudo sh")

s.execute("sudo groupadd -f docker; sudo usermod -aG docker $USER")

1. **Set up the AMD GPU**

s.execute("sudo apt update; wget https://repo.radeon.com/amdgpu-install/6.3.3/ubuntu/noble/amdgpu-install\_6.3.60303-1\_all.deb")

s.execute("sudo apt -y install ./amdgpu-install\_6.3.60303-1\_all.deb; sudo apt update")

s.execute("amdgpu-install -y --usecase=dkms")

s.execute("sudo apt -y install rocm-smi")

s.execute("sudo usermod -aG video,render $USER")

s.execute("sudo reboot")

time.sleep(30)

s.refresh()

s.check\_connectivity()

s.execute("rocm-smi")

s.execute("sudo apt -y install cmake libncurses-dev libsystemd-dev libudev-dev libdrm-dev libgtest-dev")

s.execute("git clone https://github.com/Syllo/nvtop")

s.execute("mkdir -p nvtop/build && cd nvtop/build && cmake .. -DAMDGPU\_SUPPORT=ON && sudo make install")

1. **Build a container image - for MLFlow section**

s.execute("docker build -t jupyter-mlflow -f mltrain-chi/docker/Dockerfile.jupyter-torch-mlflow-rocm .")

**Set up the Nvidia GPU**

s.execute("git clone --recurse-submodules [https://github.com/Jasonzzzz28/Transformer-QA")](https://github.com/Jasonzzzz28/Transformer-QA\"))

s.execute("curl -sSL https://get.docker.com/ | sudo sh")

s.execute("sudo groupadd -f docker; sudo usermod -aG docker $USER")

s.execute("curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | sudo gpg --dearmor -o /usr/share/keyrings/nvidia-container-toolkit-keyring.gpg \

&& curl -s -L https://nvidia.github.io/libnvidia-container/stable/deb/nvidia-container-toolkit.list | \

sed 's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia-container-toolkit-keyring.gpg] https://#g' | \

sudo tee /etc/apt/sources.list.d/nvidia-container-toolkit.list")

s.execute("sudo apt update")

s.execute("sudo apt-get install -y nvidia-container-toolkit")

s.execute("sudo nvidia-ctk runtime configure --runtime=docker")

# for https://github.com/NVIDIA/nvidia-container-toolkit/issues/48

s.execute("sudo jq 'if has(\"exec-opts\") then . else . + {\"exec-opts\": [\"native.cgroupdriver=cgroupfs\"]} end' /etc/docker/daemon.json | sudo tee /etc/docker/daemon.json.tmp > /dev/null && sudo mv /etc/docker/daemon.json.tmp /etc/docker/daemon.json")

s.execute("sudo systemctl restart docker")

s.execute("sudo apt update")

s.execute("sudo apt -y install nvtop")

s.execute("docker build -t jupyter-mlflow -f Transformer-QA/docker/Dockerfile.jupyter-torch-mlflow-cuda .")

1. **prepare data**

docker volume create \

--driver local \

--opt type=none \

--opt device=/mnt/object/data \

--opt o=bind \

transformer-qa

docker ps

docker compose -f Transformer-QA/docker/docker-compose-data.yaml up -d

docker run --rm -it -v transformer-qa:/mnt/data alpine ls -l /mnt/data

1. **Start a ML FLOW tracking system**

docker compose -f Transformer-QA/docker/docker-compose-mlflow.yaml up -d

docker ps

1. **Start a Juypter Server**

docker image list

HOST\_IP=$(curl --silent http://169.254.169.254/latest/meta-data/public-ipv4)

docker run -d --rm -p 8888:8888 \

--device=/dev/kfd \

--device=/dev/dri \

--group-add video \

--group-add $(getent group render | cut -d':' -f3) \

--shm-size 16G \

-v ~/Transformer-QA/workspace\_mlflow:/home/jovyan/work/ \

-v transformer-qa:/mnt/data \

-e "MLFLOW\_TRACKING\_URI=http://${HOST\_IP}:8000/" \

-e "CUSTOM\_DATA\_PATH=/mnt/data/source\_code\_qa\_with\_summary\_formatted.json" \

--name jupyter-mldevop \

jupyter-mlflow

docker logs jupyter-mldevop

HOST\_IP=$(curl --silent http://169.254.169.254/latest/meta-data/public-ipv4)

docker run -d --rm -p 8888:8888 \

--gpus all \

--shm-size 16G \

-v ~/Transformer-QA/workspace\_mlflow:/home/jovyan/work/ \

-v transformer-qa:/mnt/object/ \

-e MLFLOW\_TRACKING\_URI="http://${HOST\_IP}:8000/" \

-e CUSTOM\_DATA\_PATH="/mnt/object" \

--name jupyter-mldevop \

jupyter-mlflow

docker logs jupyter-mldevop

## 8、Track a Pytorch experiment

### git branch

git pull origin main

### Run a non-MLFlow training job

cd ~/work

git clone https://github.com/Jasonzzzz28/Transformer-QA

cd ~/work/Transformer-QA

python3 train1.py

# 创建并编辑训练脚本

nano squad\_train.py

# 确保已设置环境变量

# 设置环境变量

export MLFLOW\_TRACKING\_URI=http://<your-server-ip>:8000

export CUSTOM\_DATA\_PATH=/mnt/data/source\_code\_qa\_with\_summary\_formatted.json

Ray Train:

docker build -t ray-rocm:2.42.1 -f Transformer-QA/docker/Dockerfile.ray-rocm .

export HOST\_IP=$(curl --silent http://169.254.169.254/latest/meta-data/public-ipv4 )

docker compose -f Transformer-QA/docker/docker-compose-ray-rocm.yaml up -d