A100 部署 DeepSeek-R1-AWQ

环境说明

主机名	ip 地址	系统	规格
yichang-10-200-3-23	10.200.3.23	ubuntu 22.04	8 卡 A100

安装驱动和桥接器

安装 gcc：

```Bash
apt-get update -y
apt install build-essential -y
```

安装驱动：

```Bash
wget https://us.download.nvidia.com/tesla/560.35.03/NVIDIA-Linux-x86_64-560.35.03.run
# 注意： 在交互式安装时要确认安装32位的兼容库，同时要Rebuild initramfs
sh NVIDIA-Linux-x86_64-560.35.03.run
```

> 需要说明的是驱动程序的版本与后续安装的桥接器的版本必须保持一致，所以在选择驱动版本时，需要确保桥接器也有对应的版本（并不是所有的驱动版本都能找到对应的桥接器的版本）

安装桥接器：

```Bash
# 确保桥接器版本与驱动版本完全一致（包括次版本）
wget https://developer.download.nvidia.cn/compute/cuda/repos/ubuntu2204/x86_64/nvidia-fabricmanager-560_560.35.03-1_amd64.deb

dpkg -i nvidia-fabricmanager-560_560.35.03-1_amd64.deb
systemctl enable nvidia-fabricmanager --now
systemctl status nvidia-fabricmanager
```

之后重启服务器，配置持久模式：

```Bash
nvidia-smi -pm 1
```

安装 docker 并配置 nvidia-runtime-toolkit

安装 docker：

```bash
# step 1: 安装必要的一些系统工具
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg

# step 2: 信任 Docker 的 GPG 公钥
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg

# Step 3: 写入软件源信息
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://mirrors.aliyun.com/docker-ce/linux/ubuntu \
  "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# Step 4: 安装Docker
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

# 安装指定版本的Docker-CE:
# Step 1: 查找Docker-CE的版本:
# apt-cache madison docker-ce
#   docker-ce | 17.03.1~ce-0~ubuntu-xenial | https://mirrors.aliyun.com/docker-ce/linux/ubuntu xenial/stable amd64 Packages
#   docker-ce | 17.03.0~ce-0~ubuntu-xenial | https://mirrors.aliyun.com/docker-ce/linux/ubuntu xenial/stable amd64 Packages
# Step 2: 安装指定版本的Docker-CE: (VERSION例如上面的17.03.1~ce-0~ubuntu-xenial)
# sudo apt-get -y install docker-ce=[VERSION]
```

配置 docker 使用 nvidia-runtime-toolkit：

```bash
curl -fsSL https://mirrors.ustc.edu.cn/libnvidia-container/gpgkey | sudo gpg --dearmor -o /usr/share/keyrings/nvidia-container-toolkit-keyring.gpg

curl -s -L https://mirrors.ustc.edu.cn/libnvidia-container/stable/deb/nvidia-container-toolkit.list | \
  sed 's#deb https://nvidia.github.io#deb [signed-by=/usr/share/keyrings/nvidia-container-toolkit-keyring.gpg] https://mirrors.ustc.edu.cn#g' | \
  sudo tee /etc/apt/sources.list.d/nvidia-container-toolkit.list

apt update -y
apt install -y nvidia-container-toolkit nvidia-container-runtime
```

在/etc/docker/daemon.json 中添加如下配置：

```Bash
{
  "default-runtime": "nvidia",
  "runtimes": {
    "nvidia": {
      "path": "/usr/bin/nvidia-container-runtime",
      "runtimeArgs": []
    }
  },
}
```

然后重启 docker：

```Bash
systemctl restart docker
```

获取镜像

生成 vllm 镜像的 dockerfile 示例如下：

```Docker
FROM docker.m.daocloud.io/nvidia/cuda:12.6.3-runtime-ubuntu22.04
RUN set -ex; \
    apt update -y; \
    apt install -y python3.10 python3-pip; \
    pip install vllm==v0.7.2
```

构建镜像：

```Docker
docker build -t vllm:v0.7.2 .
```

获取 open-webui 镜像：

```Bash
docker pull ghcr.m.daocloud.io/open-webui/open-webui:main
```

获取模型

deepseek-r1-awq 是 671b 的参数，但是精度只有 int4，单台 8 卡 A100 可以完成部署。

下载模型：

```Bash
mkdir /data
cd /data
git lfs install
git clone https://www.modelscope.cn/cognitivecomputations/DeepSeek-R1-awq.git
```

部署

部署 deepseek：

```Bash
docker run -d --runtime nvidia \
--gpus all \
-v /data:/mnt/models \
-p 12345:12345 \
--ipc=host \
hub.wanjiedata.com/models/vllm:v0.7.2 \
python3 -m vllm.entrypoints.openai.api_server --host 0.0.0.0 --port 12345 --max-model-len 65536 --trust-remote-code --tensor-parallel-size 8 --quantization moe_wna16 --gpu-memory-utilization 0.97 --kv-cache-dtype fp8_e5m2 --calculate-kv-scales --served-model-name deepseek-reasoner --model /mnt/models/DeepSeek-R1-AWQ
```

部署 open-webui：

```Bash
docker run -d -p 3030:8080 \
-e ENABLE_OLLAMA_API=false \
-e OPENAI_API_KEY=NULL \
-e OPENAI_API_BASE_URL=http://10.200.3.23:12345/v1 \
-e ENABLE_RAG_WEB_LOADER_SSL_VERIFICATION=false \
-v open-webui:/app/backend/data \
--name open-webui \
--restart always ghcr.m.daocloud.io/open-webui/open-webui:main
```