部署Perplexity r1-1776模型指南

前置条件

- 为了部署相关模型，您需要开通相关的ECS机型购买选项，并申请到相关的资源quota配置
- 完成VPC，EIP等服务开通和配置

部署步骤

1. 创建GPU ECS实例

1. 选择合适的region/az，以及相应的机型，配置合适大小的云盘空间

模型名称	参数量	云盘大小推荐
r1-1776	671B	建议FlexPL,2TB

2. 选择ubuntu 22.04 配置gpu driver 535版本镜像：

3. 为GPU ECS配置公网访问方式，选择绑定公网IP或者使用NAT网关均可。后续自动拉取镜像和权重需要公网访问。

2. 部署容器环境

登录2台H20 ECS 分别安装容器环境，具体安装方法参考如下：

方法一：安装nvidia-docker2

```
curl -s https://mirrors.ivolces.com/nvidia_all/ubuntu2204/x86_64/3bf863cc.pub | sudo apt-key add -
cat <<EOF >/etc/apt/sources.list.d/nvidia.list
deb http://mirrors.ivolces.com/nvidia_all/ubuntu2204/x86_64/ /
EOF
apt update
```

```
apt install nvidia-docker2
```

*方法二：**或者使用docker + nvidia container toolkit，具体如下：***

- 安装docker：

```
# Update the apt package index and install packages to allow apt to use a repository over HTTPS
sudo apt update
sudo apt install ca-certificates curl gnupg lsb-release
# Add Docker's official GPG key
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://mirrors.ivolces.com/docker/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
# Use the following command to set up the repository
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://mirrors.ivolces.com/docker/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
# update package index
sudo apt update
# Install docker-ce
sudo apt install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

- 安装nvidia-container-toolkit：

```
curl -s https://mirrors.ivolces.com/nvidia_all/ubuntu2204/x86_64/3bf863cc.pub | sudo apt-key add -
cat <<EOF >/etc/apt/sources.list.d/nvidia.list
deb http://mirrors.ivolces.com/nvidia_all/ubuntu2204/x86_64/ /
EOF
apt update
apt install nvidia-container-toolkit
sudo nvidia-ctk runtime configure --runtime=docker
sudo systemctl restart docker
```

3. 准备模型文件：

建立路径

```
# 模型权重存放路径
mkdir -p /data01/models
```

```
cd /data01/models
```

安装huggingface cli并拉取模型文件

```
# 安装HF CLI
pip install huggingface_hub[cli]
# 拉取模型文件，支持续传
huggingface-cli download perplexity-ai/r1-1776 --local-dir r1-1776
```

4. 启动模型

1. 请采用下面命令完成相关配置，启动docker和模型服务

   a. 登录节点0，执行如下docker run命令

```
## total ranks = 2
# rank 0
docker run -d --network host --privileged --gpus=all --ipc=host -v /data01:/data -v /var/run/nvidia-topologyd/:/var/run/nvidia-topologyd/ -e MODEL_NAME=r1-1776 -e MODEL_LENGTH=8192 -e TP=16 -e TOTAL_RANKS=2 -e RANKS=0 -e RANK0_ADDR=192.168.0.2:10240 -e PORT=8080 -e CMD_ARGS="--mem-fraction-static 0.95 --disable-cuda-graph" ai-containers-cn-beijing.cr.volces.com/deeplearning/sglang:0.4.2.iaas

```

   b. 登录节点1，执行如下docker run命令

```
## total ranks = 2

# rank 1cd
docker run -d --network host --privileged --gpus=all --ipc=host -v /data01:/data -v /var/run/nvidia-topologyd/:/var/run/nvidia-topologyd/ -e MODEL_NAME=r1-1776 -e MODEL_LENGTH=8192 -e TP=16 -e TOTAL_RANKS=2 -e RANKS=1 -e RANK0_ADDR=192.168.0.2:10240 -e PORT=8080 -e CMD_ARGS="--mem-fraction-static 0.95 --disable-cuda-graph" ai-containers-cn-beijing.cr.volces.com/deeplearning/sglang:0.4.2.iaas
```

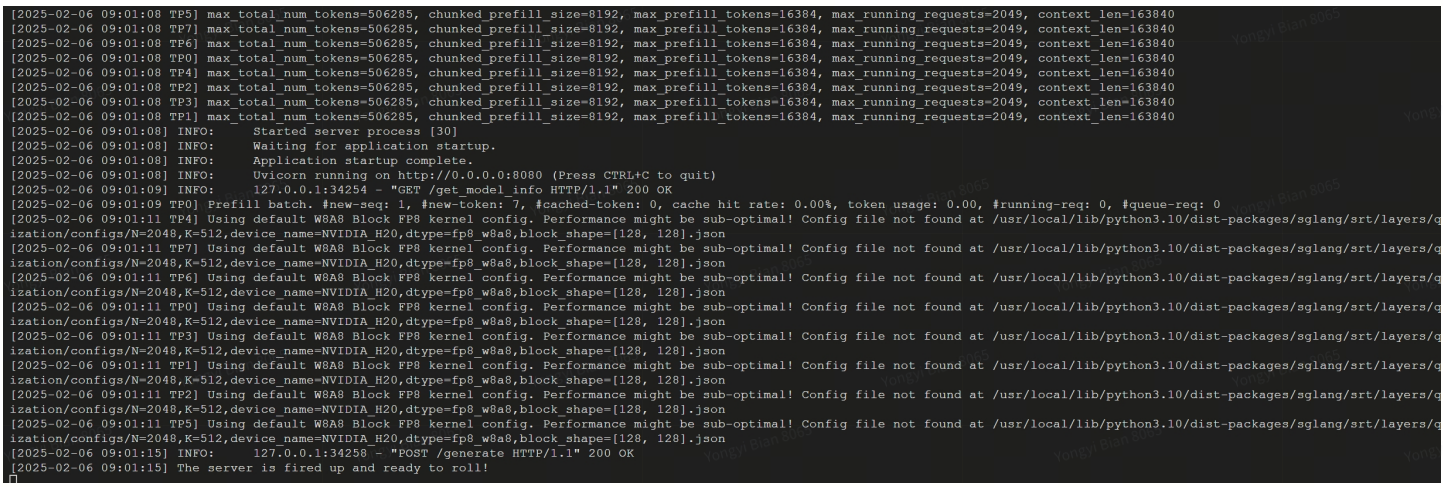
上述命令中 标黄字段的相关环境变量说明如下，需要按照实际情况修改：

环境变量	默认值	描述

MODEL_PATH	/data/models	容器内模型存储路径
MODEL_NAME	DeepSeek-R1	模型名称
MODEL_LENGTH	131072	模型的最大长度（token 数）
TP	16	Tensor Parallelism 并行度
TOTAL_RANKS	1	总节点数
RANKS	0	当前节点的rank号
RANK0_ADDR	需要指定	rank0节点的IP和端口，一般为node0 所在的内网ip
PORT	8080	服务监听的端口号

高级配置：如果要修改模型其他启动参数（如GPU_MEM_UTIL，PREFIX_CACHE等）的话，请在容器中修改模型启动脚本“/entrypoint.sh”的参数

两个节点docker分别启动以后，会自动拉取镜像和对应的权重文件，权重存放在/data/models目录下。两个节点的运行状态可以通过 docker logs查看，当节点0的docker logs显示如下时，代表模型服务已经启动成功，可以进行下一步测试操作。



2. 登录节点0，执行以下curl prompt，观察到流式生成为模型正常运行，可以进行下一步的模型调用。执行docker logs 看到模型端口是否成功启动日志。

```
curl -X POST http://0.0.0.0:8080/v1/chat/completions -H "Content-Type: application/json" -d '{
    "model": "/data/models/DeepSeek-R1",
    "messages": [
```

```
            {
                "role": "user",
                "content": "请证明一下黎曼猜想"
            }
        ],
        "stream": true,
        "max_tokens": 100,
        "temperature": 0.7
    }'
```

5. 【可选】部署NGINX

生成API Key (using python3):

```python
import secrets
api_key = secrets.token_hex(16)
print(api_key)
```

安装:

```
sudo apt update
sudo apt install nginx -y

# 创建配置文件
sudo nano /etc/nginx/sites-available/llm-proxy

```

配置文件:

```
server {
    listen 80;
    server_name [公网IP];

    set $valid_api_key 0;
    if ($http_x_api_key = "YOUR_API_KEY_1") {
        set $valid_api_key 1;
    }
    if ($http_x_api_key = "YOUR_API_KEY_2") {
        set $valid_api_key 1;
    }

```

```
    location / {
        if ($valid_api_key = 0) {
            return 403 "Forbidden: Missing or invalid API key";
        }

        proxy_pass http://[服务ip]:8080;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}

```

开启服务:

```
sudo ln -s /etc/nginx/sites-available/llm-proxy /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl restart nginx
```

排查日志:

```
# 查看日志
tail /var/log/nginx/error.log
```

测试:

```
curl -X POST http://45.78.210.224/v1/chat/completions -H "Content-Type: application/json" -H "X-API-Key: [您的API key]" -d '{
    "model": "/data/models/DeepSeek-R1",
    "messages": [
        {
            "role": "user",
            "content": "hello who are you?"
        }
    ],
    "stream": true,
    "max_tokens": 100,
    "temperature": 0.7
}'
```

```
13
```

6. 【可选】部署Streamlit简单UI

安装依赖:

```
apt install python3-venv

python3 -m venv chatbot-venv
source chatbot-venv/bin/activate

pip install streamlit requests
```

界面主文件:

```
import streamlit as st
import requests
import json

st.title("DeepSeek-R1 Chatbot")

# Initialize chat history
if "messages" not in st.session_state:
    st.session_state.messages = []

# Display chat messages from history on app rerun
for message in st.session_state.messages:
    with st.chat_message(message["role"]):
        st.markdown(message["content"])

# React to user input
if prompt := st.chat_input("What is your question?"):
    # Display user message in chat message container
    st.chat_message("user").markdown(prompt)
    # Add user message to chat history
    st.session_state.messages.append({"role": "user", "content": prompt})

    # Call the API
    response = requests.post(
        "http://45.78.228.109/v1/chat/completions",
        headers={"Content-Type": "application/json"},
        data=json.dumps({
            "model": "/data/models/DeepSeek-R1",
```

```
            "messages": st.session_state.messages
        })
    )

    if response.status_code == 200:
        assistant_response = response.json()["choices"][0]["message"]["content"]
        # Display assistant response in chat message container
        with st.chat_message("assistant"):
            st.markdown(assistant_response)
        # Add assistant response to chat history
        st.session_state.messages.append({"role": "assistant", "content": assistant_response})
    else:
        st.error(f"Error: {response.status_code} - {response.text}")

```

附录

- Dockefile中的引擎启动脚本 entrypoint.sh

```
#!/bin/bash

MODEL_PATH=${MODEL_PATH:-"/data/models"}
MODEL_NAME=${MODEL_NAME:-"DeepSeek-R1"}
MODEL_LENGTH=${MODEL_LENGTH:-131072}
TP=${TP:-8}
RANK0_ADDR=${RANK0_ADDR:-""}
RANKS=${RANKS:-0}
TOTAL_RANKS=${TOTAL_RANKS:-1}
PORT=${PORT:-8080}

# check if MODE_PATH and MODEL_NAME are set
if [ -z "$MODEL_PATH" ] || [ -z "$MODEL_NAME" ]; then
  echo "MODE_PATH and MODEL_NAME must be set"
  exit 1
fi

# check if MODE_PATH not exists, create it
if [ ! -d "$MODEL_PATH" ]; then
  mkdir -p $MODEL_PATH
fi

```

```
# check if MODE_PATH/MODEL_NAME not exists, download it
if [ ! -d "$MODEL_PATH/$MODEL_NAME" ]; then
  cd $MODEL_PATH
  oniond download model $MODEL_NAME
  if [ $? -ne 0 ]; then
    echo "Failed to download model $MODEL_NAME"
    exit 1
  fi
  cd -
fi

# check if it is multiple ranks
if [ $TOTAL_RANKS -gt 1 ]; then
    # check if RANK0_ADDR is set
    if [ -z "$RANK0_ADDR" ]; then
        echo "RANK0_ADDR must be set"
        exit 1
    fi

    GLOO_SOCKET_IFNAME=eth0 NCCL_IB_HCA=mlx5_ NCCL_IB_DISABLE=0 NCCL_SOCKET_IFNAME=eth0 NCCL_IB_GID_INDEX=3 python3 -m sglang.launch_server --model-path $MODEL_PATH/$MODEL_NAME --tp $TP --dist-init-addr $RANK0_ADDR --nnodes $TOTAL_RANKS --node-rank $RANKS --trust-remote-code --host 0.0.0.0 --port $PORT
else

    if [ $TP -gt 8 ]; then
        TP=8
    fi

    python3 -m sglang.launch_server --model-path $MODEL_PATH/$MODEL_NAME --context-length $MODEL_LENGTH --tp $TP --trust-remote-code --host 0.0.0.0 --port $PORT --mem-fraction-static 0.95

fi
```