

基于CUDA 12.9和SM_100的VLLM环境搭建

环境构建

Container

基础镜像：以 `nvcr.io/nvidia/pytorch:25.06-py3@06aa7e7a6f5a` 为基础构建容器

根据[pytorch-25.06 release info](#)

该镜像提供下列工具

Name	version
CUDA	12.9.1
Torch-TensorRT	2.8.0a0
NVIDIA DALI®	1.50
nvImageCodec	0.2.0.7
MAGMA	2.6.2
JupyterLab	4.3.6
TensorRT Model Optimizer	0.29
TransformerEngine	2.4
NVIDIA RAPIDS™	25.04
NVIDIA cuBLASmp	0.4.0

建议在此基础上构建镜像以便于提供服务

例如：

```
ARG NGC_VERSION=25.06-py3
FROM nvcr.io/nvidia/pytorch:${NGC_VERSION} AS base
ENV DEBIAN_FRONTEND=noninteractive
RUN mkdir /run/sshd && apt-get update && apt-get install -y \
    openssh-server \
    openssh-client \
    vim \
    wget \
    git
ENV TZ=Asia/Shanghai
ENV NCCL_IB_RETRY_CNT="13" NCCL_IB_TIMEOUT="22" \
    NCCL_DEBUG="WARN" \
    NCCL_IB_DISABLE="0"
RUN apt install -y net-tools sudo
RUN dpkg-statoverride --remove /usr/lib/dbus-1.0/dbus-daemon-launch-helper || true
```

继续工作

sshd

```
mkdir /run/sshd && apt-get update && apt-get install -y openssh-server
# Start by
/usr/sbin/sshd
# Or Start as a service
/usr/sbin/sshd -D
```

如果通过服务启动sshd,则可以通过下列命令管理sshd

```
# 查看状态
service ssh status
# 重启服务
service ssh restart
# 关闭服务
service ssh stop
# 启动服务
service ssh start
# 共支持方法为{start|stop|reload|force-reload|restart|try-restart|status}
```

Conda

通过miniforge来安装conda

将 [Miniforge3-24.1.2-0-Linux-x86_64.sh](#) 文件上传至容器内执行安装
也可以在[miniforge release](#)上选择其他版本

```
chmod +x ./Miniforge3-24.1.2-0-Linux-x86_64.sh
./Miniforge3-24.1.2-0-Linux-x86_64.sh
```

~/.bashrc

建议使用南科大源作为镜像源

因为该镜像源包含了pytorch和nvidia

```
channels:
  - defaults
show_channel_urls: true
default_channels:
  - https://mirrors.sustech.edu.cn/anaconda/pkg/main
  - https://mirrors.sustech.edu.cn/anaconda/pkg/free
  - https://mirrors.sustech.edu.cn/anaconda/pkg/r
  - https://mirrors.sustech.edu.cn/anaconda/pkg/pro
  - https://mirrors.sustech.edu.cn/anaconda/pkg/msys2
custom_channels:
  conda-forge: https://mirrors.sustech.edu.cn/anaconda/cloud
  msys2: https://mirrors.sustech.edu.cn/anaconda/cloud
  bioconda: https://mirrors.sustech.edu.cn/anaconda/cloud
  menpo: https://mirrors.sustech.edu.cn/anaconda/cloud
  pytorch: https://mirrors.sustech.edu.cn/anaconda/cloud
  simpleitk: https://mirrors.sustech.edu.cn/anaconda/cloud
  nvidia: https://mirrors.sustech.edu.cn/anaconda-extra/cloud
```

VLLM安装

conda环境

容器内的python构建在根目录下，建议从conda中重新构建环境

通过 `bash` 命令切换进入bash环境

执行以下命令创建并进入一个新的conda环境

其中 `${CONDA_ENV_NAME}` 为您设定的conda环境名称

```
bash
conda create -n ${CONDA_ENV_NAME} python=3.12
conda activate ${CONDA_ENV_NAME}
```

pytorch安装

使用最新的2.8.0版本torch

```
pip install \
    "numpy<2" \
    torch==2.8.0 torchvision==0.23.0 \
    --index-url https://download.pytorch.org/whl/cu129
```

VLLM和transformer安装

请通过预构建好的vllm二进制发行包来安装

```
pip install \
    "numpy<2" \
    "transformers<4.54.0" \
    vllm-0.9.3.dev0+ga5dd03c1e.d20250819.cu129-cp312-cp312-linux_x86_64.whl
```

测试

推理服务

通过下列命令启动vllm的推理服务

```
python -m vllm.entrypoints.openai.api_server \
    --served-model-name Qwen2-7B-Instruct \
    --model /root/crr/Qwen2.5-7B-Instruct \
    --gpu-memory-utilization 0.2 \
    --port 20000
```

接口测试

下列命令如果没有任何返回则证明服务已正常运行

```
curl http://127.0.0.1:20000/health
```

下列命令可以与模型进行对话

```
curl http://127.0.0.1:20000/v1/chat/completions \  
  --header 'Content-Type: application/json' \  
  --data-raw '{  
    "model": "Qwen2-7B-Instruct",  
    "messages": [{  
      "role": "user",  
      "content": "你好"  
    }],  
    "stream": false  
  }'
```