Docker y Nvidia CUDA

Antonio Mudarra Machuca

Febrero 15, 2025

Taller GDG

Quien soy

Soy Antonio Mudarra Machuca investigador en la Universidad de Jaén en el grupo de investigación SIMIDAT.

Objetivos del taller

Mostrar las capacidades de **docker** para la ejecución de modelos de IA, simplificando todo el proceso de configuración de distintos entornos de desarrollo y ejecución.

CUDA y Librerías

Instalación para Windows, accedemos a la web para estudiar como instalar el kit de desarrollo de CUDA de nvidia CUDA GUIDE, podemos descargar los drivers desde CUDA Toolkit 12.8

Con `control /name Microsoft.DeviceManager` > `Adaptadores de pantalla` podemos ver la gráfica que tiene nuestro equípo. Podemos ver todo el listado de productos de nvidia desde la web cuda-gpus.

Paquetes que incluye

- CUDA
  - CUDA Driver
  - CUDA Runtime (cudart)
  - CUDA Math Library (math.h)
- cuDNN
  - CUDA Deep Neural Network

Sistemas operativos compatibles:

- Microsoft Windows 11 24H2, 22H2-SV2, 23H2
- Microsoft Windows 10 22H2
- Microsoft Windows WSL 2
- Ubuntu 20.04, 22.04, 24.04

Descarga e instalación

Descarga e instalación para Windows CUDA Installation Guide for Microsoft Windows

Descarga e instalación para Ubuntu

NVIDIA CUDA Installation Guide for Linux

```
## Ubuntu 20.04
wget https://developer.download.nvidia.com/compute/cuda/rep
sudo mv cuda-ubuntu2004.pin /etc/apt/preferences.d/cuda-rep
wget https://developer.download.nvidia.com/compute/cuda/12.
sudo dpkg -i cuda-repo-ubuntu2004-12-8-local_12.8.0-570.86.
sudo cp /var/cuda-repo-ubuntu2004-12-8-local/cuda-*-keyring
```

```
## Ubuntu 22.04
wget https://developer.download.nvidia.com/compute/cuda/rep
sudo mv cuda-ubuntu2204.pin /etc/apt/preferences.d/cuda-rep
wget https://developer.download.nvidia.com/compute/cuda/12.
sudo dpkg -i cuda-repo-ubuntu2204-12-8-local_12.8.0-570.86.
sudo cp /var/cuda-repo-ubuntu2204-12-8-local/cuda-*-keyring
```

```
## Ubuntu 24.04
wget https://developer.download.nvidia.com/compute/cuda/rep
sudo mv cuda-ubuntu2404.pin /etc/apt/preferences.d/cuda-rep
wget https://developer.download.nvidia.com/compute/cuda/12.
sudo dpkg -i cuda-repo-ubuntu2404-12-8-local_12.8.0-570.86.
sudo cp /var/cuda-repo-ubuntu2404-12-8-local/cuda-*-keyring
```

```
## WSL 2
wget https://developer.download.nvidia.com/compute/cuda/rep
sudo mv cuda-wsl-ubuntu.pin /etc/apt/preferences.d/cuda-rep
wget https://developer.download.nvidia.com/compute/cuda/12.
sudo dpkg -i cuda-repo-wsl-ubuntu-12-8-local_12.8.0-1_amd64
sudo cp /var/cuda-repo-wsl-ubuntu-12-8-local/cuda-*-keyring
```

```
# Instalación
sudo apt-get update
sudo apt-get -y install cuda-toolkit-12-8
```

Tras la instalación es posible que se requiera un reinicio, podemos comprobar si los drivers de nvidia están instalados con el comando `nvidia-smi` (NVIDIA System Management Interface).

OLLAMA

Ollama es un gestor de modelos LLM que permite descargar, ejecutar y desplegar modelo LLM fácilmente mediante un servidor que distribuye una API.

Esta tecnologia nos permite simplicar mucho la distribución de modelos para distintos usos.

Docker

```
docker compose build runner
docker compose up runner -d
docker compose exec runner bash
docker compose exec runner python -c "import torch; print(t


#x = torch.rand(100, 100, 100, device='cuda:0');
#del x;
#torch.cuda.reset_max_memory_allocated(0);

docker compose exec runner python -c \
"import torch;
from tabulate import tabulate;
info_cuda = [
    ['torch.__version__', torch.__version__],
    ['torch cuda is_available', torch.cuda.is_available()],
    ['torch cuda current_device', torch.cuda.current_device
    ['torch cuda device_count', torch.cuda.device_count()],
```

docker compose de un chat-gpt propio

```yaml
services:
  open-webui:
    image: ghcr.io/open-webui/open-webui:main
    container_name: ${PROJECT_NAME}_open-webui
    volumes:
      - local-open-webui:/app/backend/data
    depends_on:
      - ollama
    ports:
      - ${OPEN_WEBUI_PORT-3000}:8080
    environment:
      - 'OLLAMA_BASE_URL=http://ollama:11434'
    extra_hosts:
      - host.docker.internal:host-gateway
    restart: unless-stopped

  ollama:
    image: ollama/ollama:latest
```

Traefik

Para traefik debemos añadir la redireccion al servicio, con ubuntu/debian `sudo nano /etc/hosts` y para Windows abrir editor de texto con permisos de administrador el fichero `C:\Windows\System32\drivers\etc\hosts` y añadir la

```
# Añadimos el host
127.0.0.1 chat.nonodev96.dev
```

Referencias

- CUDA GUIDE.
- cuDNN.
- OPEN WEB UI.