安裝步驟與測試說明。

1. 執行環境 : ubuntu 22.0.4 ，已經安裝 docker /docker compose
2. 資料夾檔案說明 : 將以下兩個資料夾，複製到 linux 使用者的 home目錄下

/api\_server

readme.docx 本檔案

really.wav TTS參考音檔

requirements.txt PIP install lib

web\_tts\_asr.py api\_server for ASR+TTS

ollama-api.yml ollama\_api\_server 用docker compose啟動

/client\_test 發出api request 的測試程式

car.mp3 ASR 測試檔

config.yml 測試設定檔

main 測試執行檔

performance\_mistral-nemo:latest\_yyyymmdd\_xxxx.csv 測試過後會產生的結果

1. 前置條件 : 安裝 python 虛擬環境與pip套件

conda create --name tts python=3.11.11

conda activate tts

cd /api\_server

pip install -r requirements.txt

pip install -y uvicorn

pip uninstall -y pydantic

pip install --no-cache-dir pydantic

pip install fastapi

啟動ASR/TTS api server

python -m uvicorn web\_tts\_asr:app --host 0.0.0.0 --port 8000

……

> You must confirm the following:

| > "I have purchased a commercial license from Coqui: licensing@coqui.ai"

| > "Otherwise, I agree to the terms of the non-commercial CPML: https://coqui.ai/cpml" - [y/n]

| | > y

啟動LLM ollama api server

docker compose -f ollama\_api.yml up -d

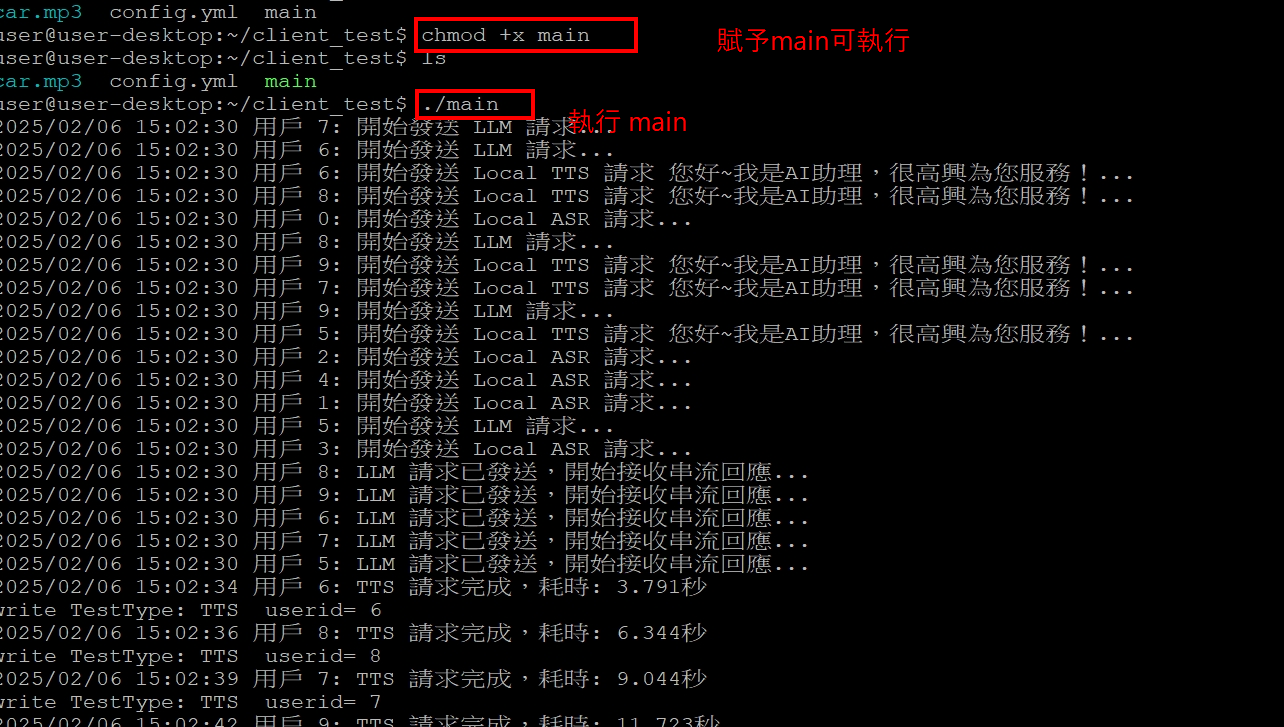
1. 進行測試

於linux 切換到　/client\_test 目錄下，

先確認 config.yml 都與安裝的server api 一致

然後執行 ./main (執行之前可能要先 chmod +x main ，修改成可執行)

正常可以看到如下的圖示

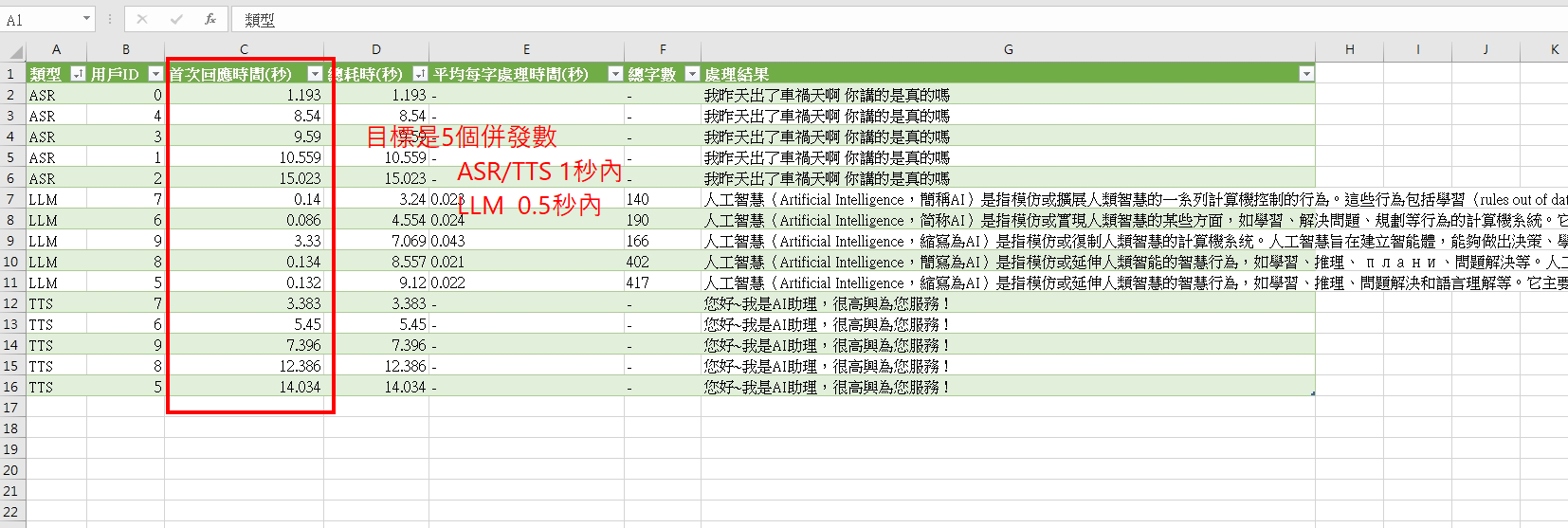


完成後可以看看有沒有產生

performance\_mistral-nemo:latest\_yyyymmdd\_xxxx.csv

1. 其他說明

測試結果 將 performance\_mistral-nemo:latest\_yyyymmdd\_xxxx.csv 轉成 xls 後排序如下 (第一會比較久，計算的是當模型已經被載入GPU後推論的時間)



ollama-api.yml 檔案說明

以下/home/jack/ollama\_models 要改成你實際環境的路徑(用來存放下載ollama的模型)

services:

  ollama:

    image: ollama/ollama:latest

    hostname: ollama

    ports:

      - "11434:11434"

    volumes:

      - /home/jack/ollama\_models:/root/.ollama/models

    networks:

      - genai-network

    deploy:

      resources:

        reservations:

          devices:

            - driver: nvidia

              count: all

              capabilities: [gpu]

networks:

  genai-network:

    driver: bridge

    name: genai-network

config.yml 檔案說明

asr:

  #url: "https://twcc-speech.thebarkingdog.tw/api/v1/asr"

  url: "http://127.0.0.1:8000/asr"

  model\_size: "medium"

  concurrent\_users: 5

  audio\_file: "car.mp3"

tts:

  url: "http://127.0.0.1:8000/tts"

  text: "您好~我是AI助理，很高興為您服務！"

  concurrent\_users: 5

llm:

  url: "http://127.0.0.1:11434/api/generate"

  model: "mistral-nemo:latest"

  prompt: "請用繁體中文回答：什麼是人工智能？"

  concurrent\_users: 5

想要加多執行緒，可以改 concurrent\_users

想要改llm 模型 ，可以改 model

想要改whispter 模型大小，可以改model\_size

如果有port 的衝突可以修改 8000 or 11434 (但server api 也要跟著改)

其他不要動。

!!!!!! 使用TTS問題 !!!!

啟動程式後可能會遇到，以下問題

*pickle.UnpicklingError: Weights only load failed. This file can still be loaded, to do so you have two options, do those steps only if you trust the source of the checkpoint.*

*(1) In PyTorch 2.6, we changed the default value of the `weights\_only` argument in `torch.load` from `False` to `True`. Re-running `torch.load` with `weights\_only` set to `False` will likely succeed, but it can result in arbitrary code execution. Do it only if you got the file from a trusted source.*

*(2) Alternatively, to load with `weights\_only=True` please check the recommended steps in the following error message.*

*WeightsUnpickler error: Unsupported global: GLOBAL TTS.tts.configs.xtts\_config.XttsConfig was not an allowed global by default. Please use `torch.serialization.add\_safe\_globals([XttsConfig])` or the `torch.serialization.safe\_globals([XttsConfig])` context manager to allowlist this global if you trust this class/function.*

解決方式 : 編輯以下檔案

/home/*使用者*/anaconda3/envs/tts/lib/python3.11/site-packages/TTS/utils/io.py

加入 ,weights\_only=False (參考下圖)

