⚛ llama.cpp

Run high-performance GGUF inference with CUDA 13.2 using llama.cpp and the Docker image `sandichhuu/llama-cpp:cu132`.

This setup is optimized for:

- Large GGUF models
- Long context inference
- Multi-modal models with `mmproj`
- NVIDIA GPU acceleration
- Speculative decoding (`draft-mtp`)
- High-throughput inference workloads

Features

- CUDA 13.2 support
- Prebuilt `llama-server`
- GGUF + multimodal support
- Optimized KV cache quantization
- Flash Attention enabled
- Long context support (`262144`)
- NVIDIA Container Toolkit compatible
- Ready for Docker Compose deployment

Requirements

Before starting, make sure you have:

- Docker
- Docker Compose
- NVIDIA Driver installed
- NVIDIA Container Toolkit

Verify GPU access:

```
docker run --rm --gpus all nvidia/cuda:13.2.0-base-ubuntu24.04 nvidia-smi
```

Docker Compose Example

```
services:
  llama-cpp:
    image: sandichhuu/llama-cpp:cu132
    container_name: llama-cpp

    ports:
      - 8080:8080

    environment:
      - GGML_CUDA_GRAPH_OPT=1
      - LD_LIBRARY_PATH=/usr/local/nvidia/lib64:/usr/local/nvidia/lib:/usr/lib/x86_64-linux-
gnu

    privileged: true
    ipc: host

    volumes:
      - ./models:/app/models

    command: >
      -m
      /app/models/unsloth/Qwen3.6-35B-A3B-MTP-GGUF/Qwen3.6-35B-A3B-UD-IQ4_NL.gguf
      --mmproj /app/models/unsloth/Qwen3.6-35B-A3B-MTP-GGUF/mmproj-BF16.gguf
      --port 8080
      --host 0.0.0.0
      -t 12
      -c 262144
      --parallel 1
      -fa on
      --cache-type-k q4_0
      --cache-type-v q4_0
      --n-cpu-moe 35
      --no-context-shift
      -b 4096
      -ub 2048
      --no-mmap
      --direct-io
      --fit off
      --jinja
      --no-cache-prompt
      --cache-ram 0
      --spec-type draft-mtp
      --spec-draft-n-max 2

    deploy:
      resources:
        reservations:
          devices:
            - driver: nvidia
              count: all
              capabilities:
                - gpu

    restart: unless-stopped
```

Folder Structure

```
.
??? docker-compose.yml
??? models
    ??? unsloth
        ??? Qwen3.6-35B-A3B-MTP-GGUF
```

Start the Container

```
docker compose up -d
```

Check logs:

```
docker logs -f llama-cpp
```

OpenAI-Compatible API

The server exposes an OpenAI-compatible API on:

```
http://localhost:8080
```

Example request:

```
curl http://localhost:8080/v1/chat/completions \
  -H "Content-Type: application/json" \
  -d '{
    "model": "Qwen",
    "messages": [
      {
        "role": "user",
        "content": "Hello"
      }
    ]
  }'
```

Explanation of Important Flags

Flag	Description
`-fa on`	Enables Flash Attention

Flag	Description
`-c 262144`	Sets context length to 262k
`--cache-type-k q4_0`	Quantized KV cache for lower VRAM usage
`--cache-type-v q4_0`	Quantized V cache
`--no-mmap`	Fully loads model into memory
`--direct-io`	Reduces page cache overhead
`--parallel 1`	Single inference stream
`--spec-type draft-mtp`	Enables speculative decoding
`--spec-draft-n-max 2`	Number of speculative draft tokens
`--mmproj`	Loads multimodal projection model
`--jinja`	Enables chat template rendering

Performance Notes

This configuration is tuned for large-scale inference workloads:

- Better throughput on large context windows
- Reduced VRAM usage using quantized KV cache
- Lower latency with CUDA graph optimization
- Improved token generation using speculative decoding
- Optimized for high-end NVIDIA GPUs

Recommended GPUs:

- RTX 4090
- RTX 5090
- A100
- H100
- L40S

Model Sources

You can download GGUF models from:

- https://huggingface.co/models?library=gguf
- https://huggingface.co/unsloth

Docker Image

Docker Hub:

Link DockerHub

Pull manually:

```
docker pull sandichhuu/llama-cpp:cu132
```

Revision #6
Created 2026-05-24 01:51:47 UTC by sandichhuu
Updated 2026-05-28 06:03:34 UTC by sandichhuu