Install SGLang

Print to PDF

Contents

You can install SGLang using any of the methods below.

Method 1: With pip

```
pip install --upgrade pip
pip install "sglang[all]" --find-links https://flashinfer.ai/whl/cu124/torch2.4/flashinfer/
```

Note: Please check the FlashInfer installation doc to install the proper version according to your PyTorch and CUDA versions.

Method 2: From source

```
# Use the last release branch
git clone -b v0.4.1.post7 https://github.com/sgl-project/sglang.git
cd sglang

pip install --upgrade pip
pip install -e "python[all]" --find-links https://flashinfer.ai/whl/cu124/torch2.4/flashinfer/
```

Note: Please check the FlashInfer installation doc to install the proper version according to your PyTorch and CUDA versions.

Note: To AMD ROCm system with Instinct/MI GPUs, do following instead:

```
# Use the last release branch
git clone -b v0.4.1.post7 https://github.com/sgl-project/sglang.git
cd sglang

pip install --upgrade pip
pip install -e "python[all_hip]"
```

Back to top

Method 3: Using docker

The docker images are available on Docker Hub as lmsysorg/sglang, built from Dockerfile. Replace `<secret>` below with your huggingface hub token.

```
docker run --gpus all \
    --shm-size 32g \
    -p 30000:30000 \
    -v ~/.cache/huggingface:/root/.cache/huggingface \
    --env "HF_TOKEN=<secret>" \
    --ipc=host \
    lmsysorg/sglang:latest \
    python3 -m sglang.launch_server --model-path meta-llama/Llama-3.1-8B-Instruct --host 0.0.0.0 --port 30
```

Note: To AMD ROCm system with Instinct/MI GPUs, it is recommended to use `docker/Dockerfile.rocm` to build images, example and usage as below:

```
docker build --build-arg SGL_BRANCH=v0.4.1.post7 -t v0.4.1.post7-rocm620 -f Dockerfile.rocm .

alias drun='docker run -it --rm --network=host --device=/dev/kfd --device=/dev/dri --ipc=host \
    --shm-size 16G --group-add video --cap-add=SYS_PTRACE --security-opt seccomp=unconfined \
    -v $HOME/dockerx:/dockerx -v /data:/data'

drun -p 30000:30000 \
    -v ~/.cache/huggingface:/root/.cache/huggingface \
    --env "HF_TOKEN=<secret>" \
    v0.4.1.post7-rocm620 \
    python3 -m sglang.launch_server --model-path meta-llama/Llama-3.1-8B-Instruct --host 0.0.0.0 --port 30

# Till flashinfer backend available, --attention-backend triton --sampling-backend pytorch are set by defa
drun v0.4.1.post7-rocm620 python3 -m sglang.bench_one_batch --batch-size 32 --input 1024 --output 128 --mo
```

Method 4: Using docker compose

▶ More

Method 5: Run on Kubernetes or Clouds with SkyPilot

▶ More

Common Notes

- FlashInfer is the default attention kernel backend. It only supports sm75 and above. If you encounter any FlashInfer-related issues on sm75+ devices (e.g., T4, A10, A100, L4, L40S, H100), please switch to other kernels by adding `--attention-backend triton --sampling-backend pytorch` and open an issue on GitHub.
- If you only need to use OpenAI models with the frontend language, you can avoid installing other dependencies by using `pip install "sglang[openai]"`.
- The language frontend operates independently of the backend runtime. You can install the frontend locally without needing a GPU, while the backend can be set up on a GPU-enabled machine. To install the frontend, run `pip install sglang`, and

for the backend, use `pip install sglang[srt]`. This allows you to build SGLang programs locally and execute them by connecting to the remote backend.