AI Workshop – LLM Inference

Github Repo: https://github.com/GCAP-Private/ai_workshop

GPU Resources

sh03-18n07: 8 A100/80G GPU, 128 CPU cores, 1000G Memory

sh04-06n05: 4 H100/80G GPU, 64 CPU cores, 1000G Memory

marlowe: 31 nodes, 8 H100/80G GPU per node

Open Source LLMs

Meta Model - Llama 3.3 70B

Google Model - gemma3 27B

OpenAI Models - GPT OSS 120B, GPT OSS 20B

Qwen Models - Qwen 2.5 72B, Qwen 3 235B

Location:

```
/oak/stanford/groups/maggiori/GCAP/data/llm_models/
```

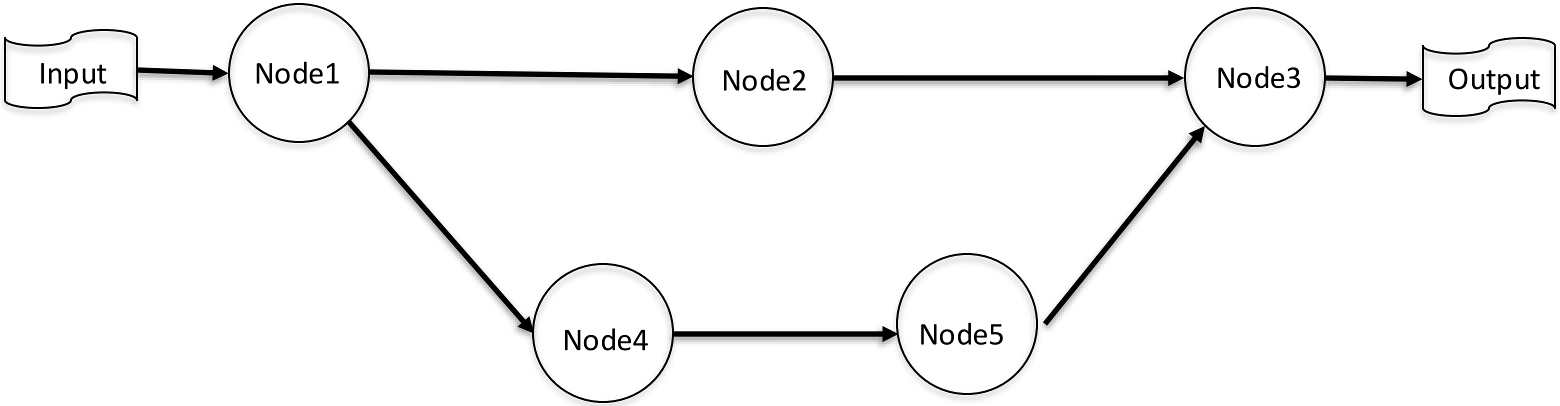
Download Models from Huggingface

Example:

```
source activate gpu1
cd $gcap_data/llm_models/qwen3
mkdir -p Qwen3-30B-A3B
huggingface-cli download Qwen/Qwen3-30B-A3B --local-dir ./Qwen3-30B-A3B
```

AI1 Architecture

vLLM-powered DAG (Directed Acyclic Graph) workflow framework for processing financial documents and transcripts



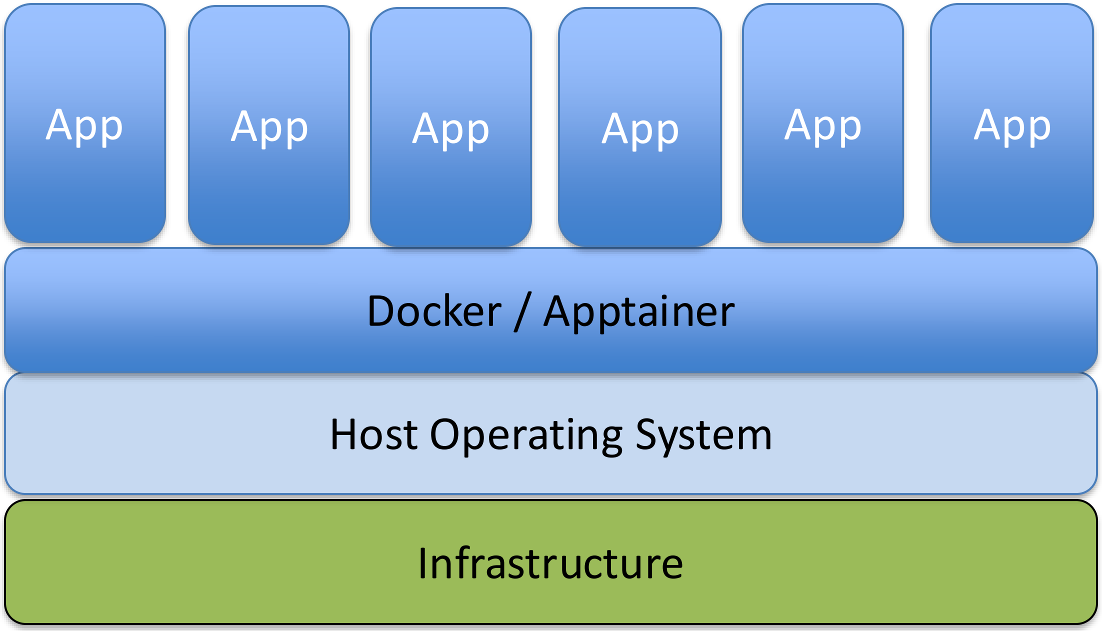
DAG (Directed Acyclic Graph)

Python Project Setup

```
my_project/
│
├── src/                    # Source code (alternative: project_name/)
│   └── my_package/
│       ├── __init__.py
│       ├── main.py
│       └── module1.py
│
├── docs/                   # Documentation
│   └── README.md
│
├── .gitignore              # Git ignore file
├── README.md               # Project description
├── requirements.txt        # Dependencies
├── setup.py                # Package installation script
├── pyproject.toml          # Modern Python project config
├── Makefile        # Build and run tasks automation
```

Container

Standard unit of software that packages up code and all its dependencies, so the application runs quickly and reliable from one computing environment to anotherl



Build and Run AI1 Pipeline

Python project task automation with Makefile

Build targets:

- build_ai1
- build_vllmapi

Run targets:

- sync_prompts
- run_* targets
- run_vllm_api

Steps to Process Transcripts

Load transcripts

Configure DAG YAML file

Make sure correct prompts are in the right place

Submit SLURM job

Monitor progress

Same steps to process jpm reports, orbitfin transcripts, fitch reports

Run LLM as API

```
make run_vllm_api
```

Example:

```
ai_workshop/llm_inference/vllmapi_transcript_analysis.ipynb
```

LLM Application - RAG

Retrieval Augmented Generation

Semantic search with embeddings

LLM entity extraction

Example:

ai_workshop/llm_inference/rag_search.ipynb