MLFO Demonstration using Reference Implementation

Abhishek Dandekar < dandekar@campus.tu-berlin.de >

Background

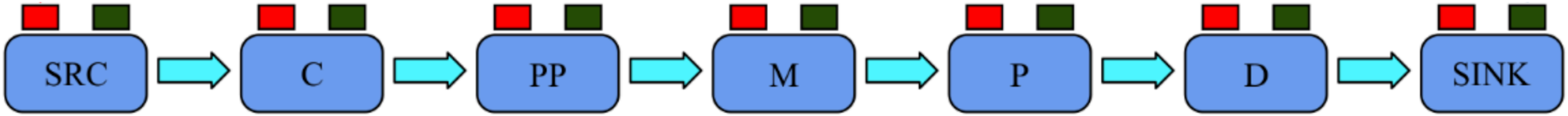
ITU Y.3172

Overview

- This standard uses a set of logical nodes(ML pipeline) which are combined to form a machine learning application
- Pipeline nodes include source, collector, preprocessor, model, policy, distributor and sink
- ML pipeline is managed and orchestrated by a entity called MLFO (Machine learning function orchestrator)
- This pipeline is orchestrated based on intent from the user

Source: ITU Y.3172 https://www.itu.int/rec/T-REC-Y.3172-201906-I/en

ML Pipeline



Source: ITU Y.3172 https://www.itu.int/rec/T-REC-Y.3172-201906-I/en

Intent

- Intent is a high level description of the ML application by the user
- It may consist of:
  - Description of pipeline nodes (e.g source, sink, model etc.)
  - Constraints (e.g cpu, gpu, memory constraints, use case constraints etc.)

ML marketplace

- ML marketplace can be-
  - Internal
  - External (e.g Acumos)
- MLFO fetches the correct model from ML marketplace based on intent

Orchestration

- MLFO talks with underlying NFV orchestrator (e.g Kubernetes, OSM) to orchestrate the pipeline
- e.g MLFO may talk with orchestrator in order to perform cluster selection for-
    - Training
    - Inference

Solution

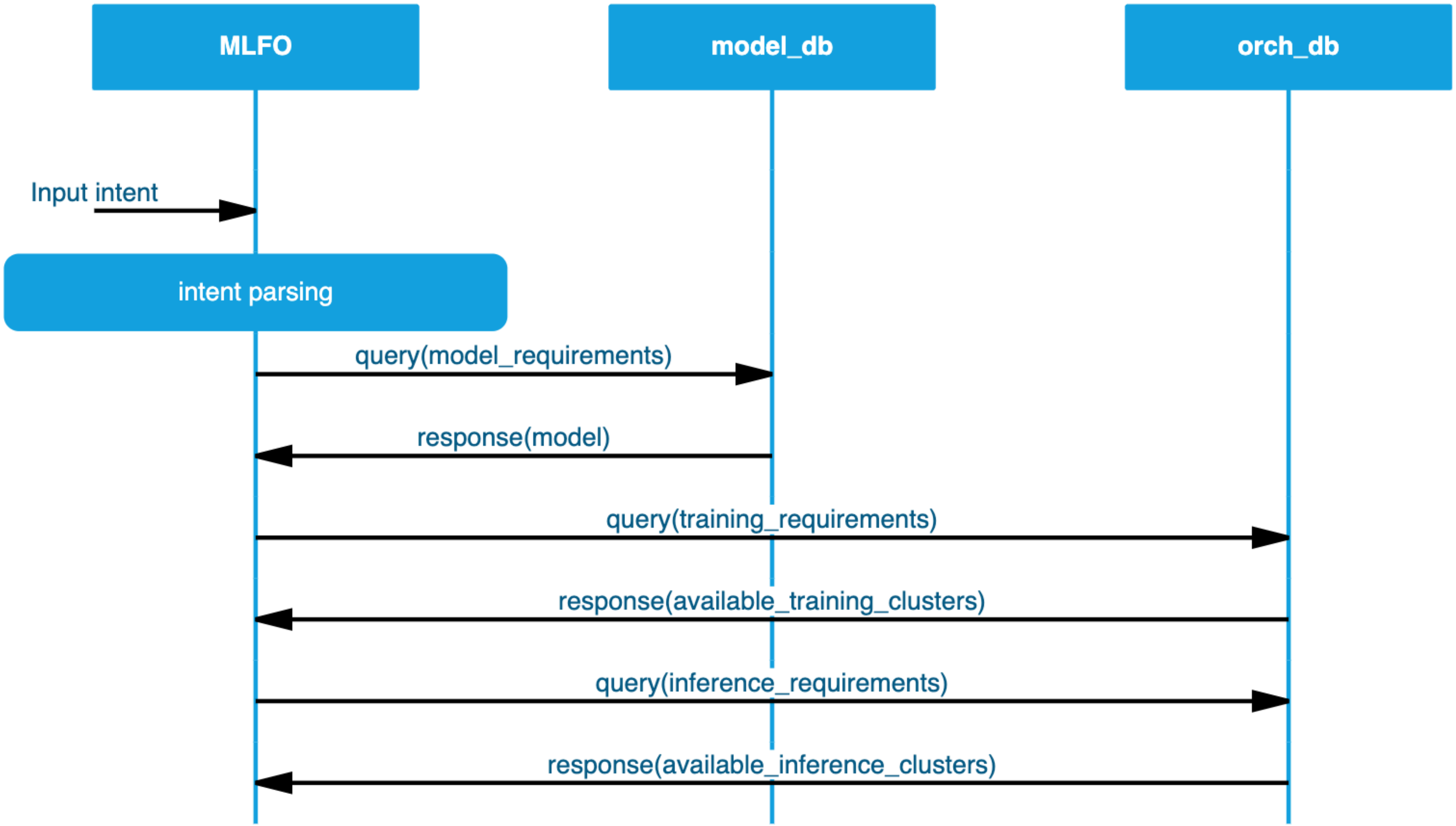
Functional Overview

- The solution consists of implementation of minimal version of MLFO
- MLFO performs model selection based on use case specified in intent
  - e.g for edge use case, select a private model with low resourceRequirements
- It fetches the correct model from the Marketplace (model_db)
- Based on the model requirements (number of GPUs required) it queries the orchestrator database (orch_db) for available training clusters

Functional Overview

- Based on the requirements, the database returns correct cluster ID
    - e.g for a lighter model, a small cluster with 5 GPUs
- Similarly MLFO queries the orchestration database for inference clusters based on required number of GPUs

Flow diagram


MLFO
model_db
orch_db
Input intent
intent parsing
query(model_requirements)
response(model)
query(training_requirements)
response(available_training_clusters)
query(inference_requirements)
response(available_inference_clusters)

Implementation

Overview

- This application uses Golang, MySQL, YAML for the application and docker for containerisation
- The app consists of three main components
    - MLFO
    - Intent
    - Database

Implementation

MLFO

- *mlfo.go* contains the main code for MLFO
- It is responsible for-
  - Intent parsing
  - Model selection
  - Fetching model from External Marketplace
  - Training cluster selection
  - Inference cluster selection

Implementation

Intent

- This app uses two example intents to demonstrate two different use cases
  - edge_intent
  - cloud_intent
- Both intents contain specifications for the pipeline nodes which need to be orchestrated
- The application uses YAML for intent serialisation

Implementation

Database

- Database consists of three tables
  - models
    - Emulates External Marketplace
  - trainingClusters
    - Emulates orchestrator db
  - inferenceClusters
    - Emulates orchestrator db

Database table examples

id	uri	accessType	trainingTime	nGPU	resourceReq
model.edgeML	http://get.model.edgeML	private	low	5	low
model.cloudML	http://get.model.cloudML	public	high	10	high

models

id	gpuPresent	nGPU
cluster.training.large	1	10
cluster.training.small	1	5

trainingClusters

id	gpuPresent	nGPU
cluster.inference.small	1	1
cluster.inference.large	1	2

inferenceClusters

How do I run the app?

- Clone the following repo from github-
    - https://github.com/ITU-AI-ML-in-5G-Challenge/MLFO-Solution_xx01
- Follow the README !!!