MLFO Demonstration using Reference Implementation

Final presentation

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

15 December 2020

Background

Machine learning function orchestrator

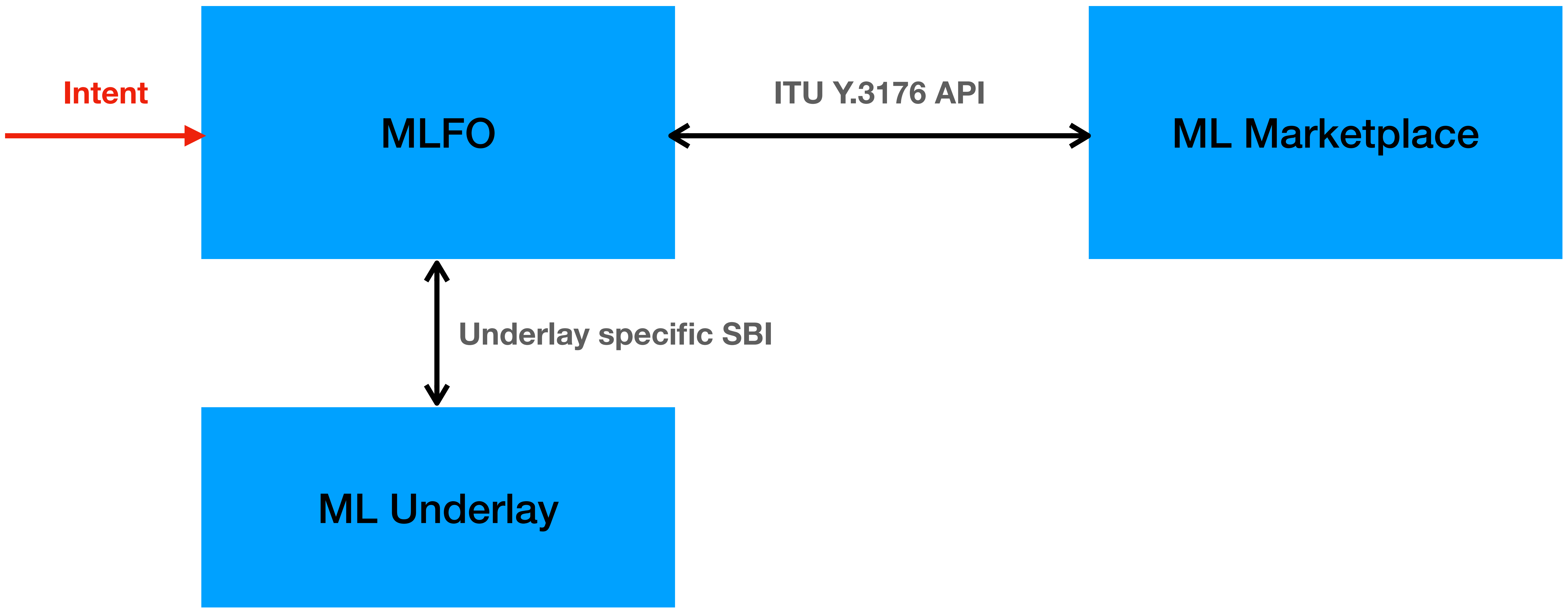
Overview

- Defined in ITU-T Y.3172
- Orchestrates ML pipelines in the network by interacting with ML underlay
- ML use cases are fed to MLFO via intent
- Based on intent MLFO performs-
  - Model management
  - Data management
  - Pipeline management

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

Machine learning function orchestrator

Figure



Challenge

Demonstrate following MLFO capabilities

- Handle input intent
- Model selection
- Fetch model from ML Marketplace
- Interact with ML underlay

Solution

A minimal implementation of MLFO using open-source components

Intent Handling

- Operator can specify input intent in a yaml
- Intent may describe:
  - Operator use case (e.g edge/cloud/distributed)
  - Target application which uses the generated insights (e.g RAN-CU)
- Based on the intent, model requirements are generated

Model selection/Model fetch

- ML marketplace is queried with generated model requirements
  - Ideally Acumos, currently using custom db
- If a matching model is found, it is fetched from the ML marketplace
- This is aligned with API described in ITU-T Y.3176

Source: ITU-T Y.3176 https://www.itu.int/rec/T-REC-Y.3176-202009-P

Interaction with ML underlay

- MLFO queries the ML underlay about availability of following resources:
    - Number of clusters available for training
        - nGPUs in training cluster
    - Number of clusters available for inference
        - nGPUs in inference cluster

Additional features

Intelligent resource allocation algorithm

- MLFO can have a global overview of GPU resources across network
- It can allocate set of GPUs for a given ML job by factoring-
    - Location of source / sink nodes
    - Network characteristics e.g Latency/Bandwidth
    - Model resource requirements for training/inference
    - Policy set by the operator

Future extensions

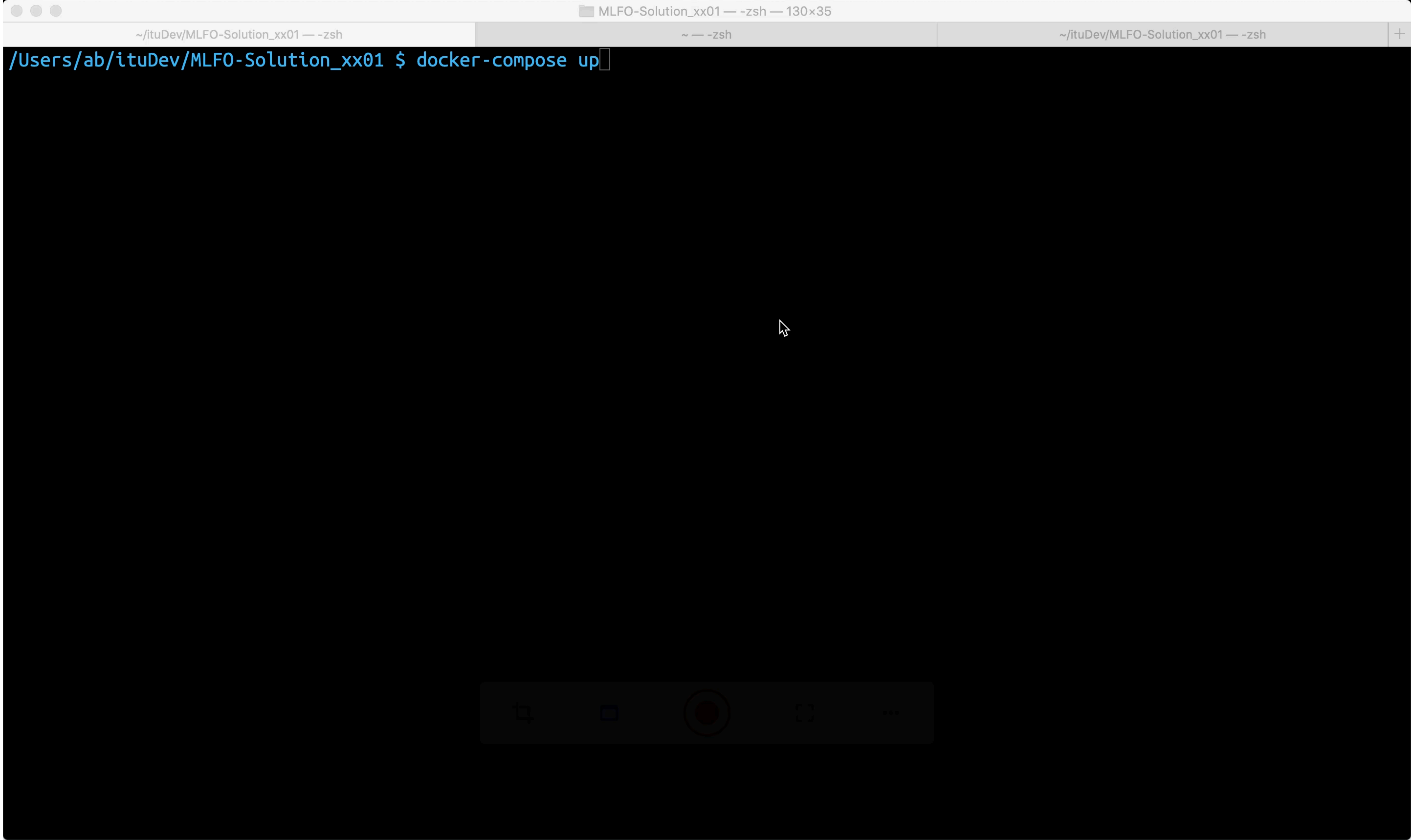
- MLFO can be integrated with-
  - ONAP DCAE to centrally orchestrate ML apps
  - ORAN Non-RT RIC to orchestrate ML apps in RAN
- *Smart* MLFO-
  - It can adapt its functionalities according to the dynamic network
- MLFO could be extended to use in hierarchical fashion for granular control

Host Feedback Incorporation

- Add references to ITU specifications
  - Added references to ITU Y.3176
- Add new scenarios and future work
  - Added new scenarios and future work

Thanks!

Questions?


MLFO-Solution_xx01 — -zsh — 130×35
~/ituDev/MLFO-Solution_xx01 — -zsh
~ — -zsh
~/ituDev/MLFO-Solution_xx01 — -zsh

```
/Users/ab/ituDev/MLFO-Solution_xx01 $ docker-compose up
```