KubeEdge AI

Motivation

Currently, "Edge AI" in the industry is at an early stage of training on the cloud and inference on the edge. However, the future trend has emerged, and related research and practice are booming, bringing new value growth points for edge computing and AI. Also, edge AI applications have much room for optimization in terms of cost, model effect, and privacy protection.

This proposal provides a basic framework for edge-cloud collaborative training and inference, so that AI applications running at the edge can benefit from cost reduction, model performance improvement, and data privacy protection.

Goals

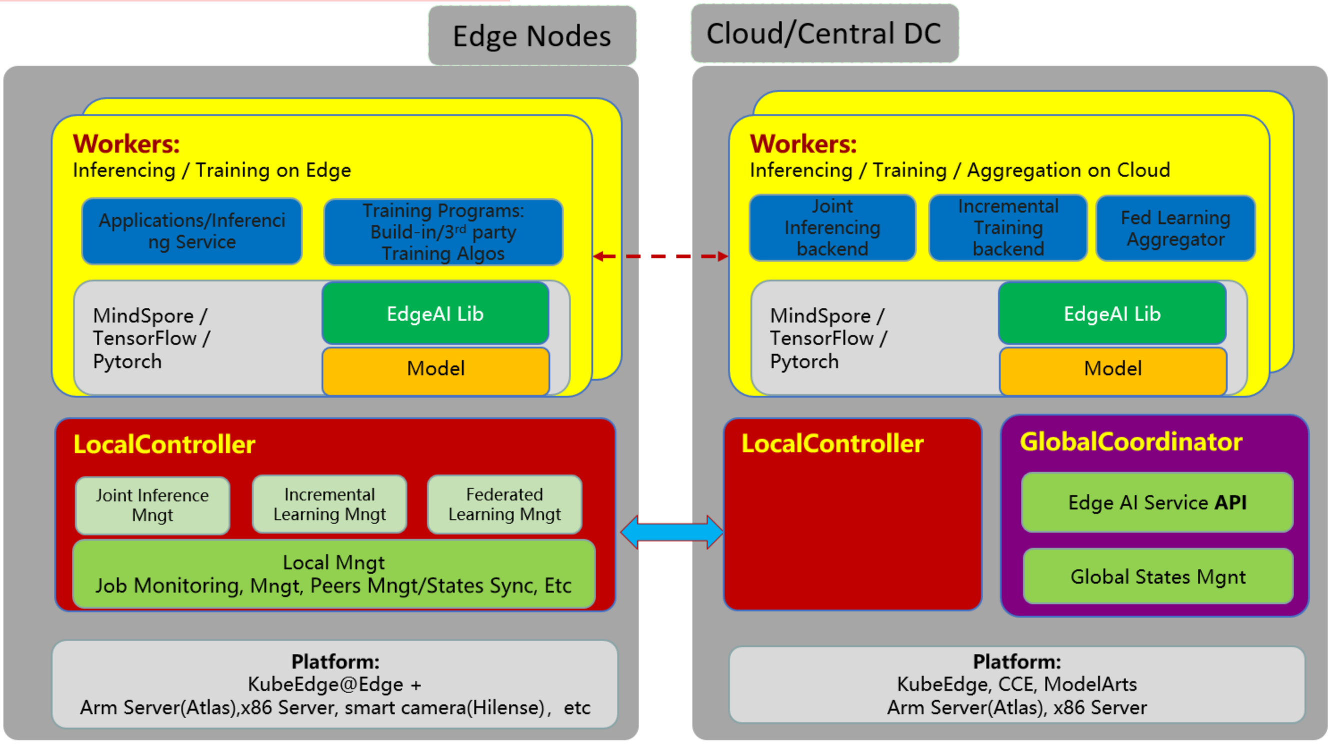
For AI applications running at the edge, the goals of edge cloud collaborative framework are:

- reducing resource cost on the edge
- improving model performance
- protecting data privacy

Proposal

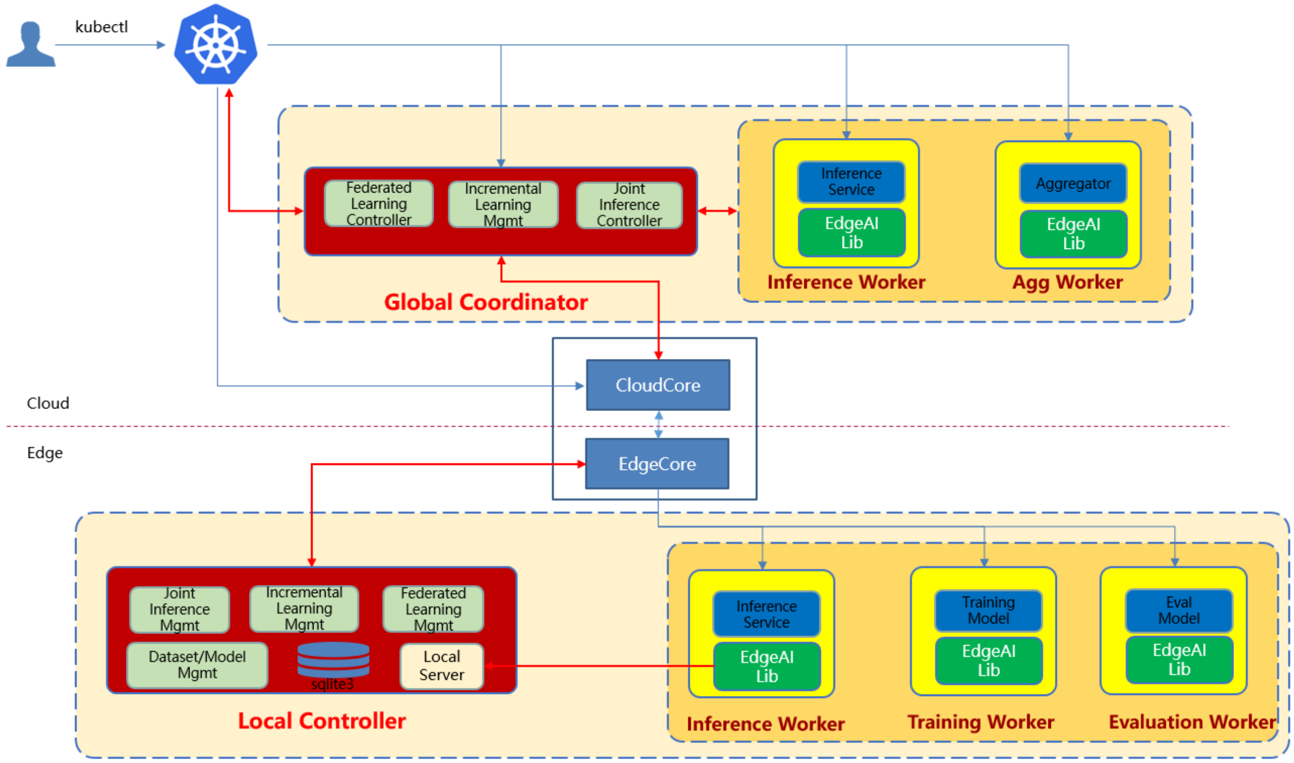
- What we propose:
    - an edge-cloud collaborative AI framework based on KubeEdge
    - with embed collaborative training and joint inferencing algorithm
    - working with existing AI framework like Tensorflow, etc
- 3 Features
    - joint inference
    - incremental learning
    - federated learning
- Targeting Users
    - Domain-specific AI Developers: build and publish edge-cloud collaborative AI services/functions easily
    - Application Developers: use edge-cloud collaborative AI capabilities.
- We are NOT:
    - to re-invent existing ML framework, i.e., tensorflow, pytorch, mindspore, etc.
    - to re-invent existing edge platform, i.e., kubeedge, etc.
    - to offer domain/application-specific algorithms, i.e., facial recognition, text classification, etc.

Broken image Broken image


Edge Nodes
Cloud/Central DC
Workers:
Inferencing / Training on Edge
Applications/Inferencing Service
Training Programs: Build-in/3rd party Training Algos
MindSpore / TensorFlow / Pytorch
EdgeAI Lib
Model
Workers:
Inferencing / Training / Aggregation on Cloud
Joint Inferencing backend
Incremental Training backend
Fed Learning Aggregator
MindSpore / TensorFlow / Pytorch
EdgeAI Lib
Model
LocalController
Joint Inference Mngt
Incremental Learning Mngt
Federated Learning Mngt
Local Mngt
Job Monitoring, Mngt, Peers Mngt/States Sync, Etc
LocalController
GlobalCoordinator
Edge AI Service API
Global States Mgnt
Platform:
KubeEdge@Edge +
Arm Server(Atlas),x86 Server, smart camera(Hilense)，etc
Platform:
KubeEdge, CCE, ModelArts
Arm Server(Atlas), x86 Server

Design Details

Architecture



- GlobalCoordinator: implements the Edge AI features controllers based on the k8s operator pattern

    - Federated Learning Controller: Implements the federated learning feature based on user created CRDs
    - Incremental Learning Controller: Implements the incremental learning feature based on user created CRDs
    - Joint Inference Controller: Implements the joint inference feature based on user created CRDs
- LocalController: manages the Edge AI features, the extra dataset/model resources on the edge nodes
- Workers: includes the training/evaluation/inference/aggregator
    - do inference or training, based on existing ML framework
    - launch on demand, imagine they are docker containers
    - different workers for different features
    - could run on edge or cloud
- Lib: exposes the Edge AI features to applications, i.e. training or inference programs