Edge Cloud-Native Cluster

› Introduction
  › Name: Adarsh Pal Singh
  › Location: Hyderabad, India
  › University: International Institute of Information Technology, Hyderabad
  › Mentor: Wenjing Chu
  › OPNFV Project: Edge Cloud and Clover Project
Edge Cloud-Native Cluster

› Project Overview
   To implement a kubernetes-based edge cluster supporting cloud-native framework and develop exemplar microservice-oriented applications for the edge as well as the edge-cloud paradigm of the future.

› Frameworks/Tools/Languages Used
   Ansible, Docker, Kubernetes, UV4L, YOLOv3, Python, Bash scripting
Edge Cloud-Native Cluster

› Project Objectives: The project had 3 major components-
  › Obj 1: Creation of a Raspberry Pi-based kubernetes edge cluster.

  › Obj 2: Implementation of a low-latency real-time microservice-based video streaming app for this edge cluster.

  › Obj 3: Edge of Tomorrow: Implementation of an Edge-Cloud based real-time object detection app employing collaborative machine learning.
Edge Cloud-Native Cluster

› Project Deliverables

› Deliverable 1: *A versatile Ansible script that can form a k8s cluster with 2 or more Raspberry Pis.*

› Deliverable 2: *Containerized bash script employing UV4L along with a k8s deployment yaml.*

› Deliverable 3: *Collaborative ML paradigm for Edge-Cloud, YOLOv3 based object detection containers for edge & GKE and respective k8s deployments. (Let’s check out this topic in detail followed by a demo!)*
Edge Cloud-Native Cluster

› Edge of Tomorrow
  › AI delivered through a cloud today

A camera
A Raspberry Pi
An actuator to control physical environment

A storage service
A server
An inference engine + application
A deep training engine

https://
Edge Cloud-Native Cluster

› Edge of Tomorrow

› Disadvantages of the current system?

A camera  A Raspberry Pi

>100’s ms, 100’s MB/s, loss of privacy ...

An actuator to control physical environment

https://

CMD

Another >100’s ms, loss of security, reliability ...

A storage service  A server  A deep training engine

An inference engine + application
Edge Cloud-Native Cluster

› Edge of Tomorrow
  › AI delivered through an Edge-Cloud of tomorrow

- A camera
- A Raspberry Pi
- An actuator to control physical environment
- Local: <5-20ms
- Local: OK MB/s
- Improved privacy
- Improved reliability, security
Edge Cloud-Native Cluster

› Edge of Tomorrow
  › AI delivered through an Edge-Cloud of tomorrow

A camera
A Raspberry Pi
An image processor
An inference engine

An actuator to control physical environment

A model serving service
Edge Cloud-Native Cluster

› Edge of Tomorrow
  › AI delivered through an Edge-Cloud of tomorrow
Edge Cloud-Native Cluster

› Edge of Tomorrow
  › AI delivered through an Edge-Cloud of tomorrow

A camera
A Raspberry Pi
An actuator to control physical environment

https://
An image processor
An inference engine

The default path
A server
A storage service
A deep training engine
A conventional app
A model serving service

THE LINUX FOUNDATION
The Linux Foundation Internal Use Only
Edge Cloud-Native Cluster

Edge of Tomorrow

AI delivered through an Edge-Cloud of tomorrow

A camera

A Raspberry Pi

An actuator to control physical environment

An image processor

An inference engine

An model serving service

A storage service

A deep training engine

A conventional app

A server

Data center

Device

Edge

Less MB/s, save cost
OK 100’s ms latency

Local: <5-20ms
Local: OK MB/s
Improved privacy, reliability, security

Improved privacy, reliability, security

Less MB/s, save cost
OK 100’s ms latency
Edge Cloud-Native Cluster

› Edge of Tomorrow
  › Demo: To trigger an action locally based on the detection of an object of interest
Edge Cloud-Native Cluster

› Edge of Tomorrow
  › Details of the demo
    › YOLOv3-Tiny Object Detection pod runs on Edge/GKE. Darknet compiled with NNPACK and ARM Neon on Edge and CUDA/cuDNN on GKE.

    › Source RPi has the 2 pods: (a) Video streaming and (b) CMD actuation from Edge/Cloud.

    › Edge/Cloud pod workflow: Capture image stream -> Run NN -> Send CMD.

    › Simple Socket programming used for sending/receiving CMD.

    › Latency test: Edge vs. Cloud!
Edge Cloud-Native Cluster

Edge of Tomorrow

A not-so-professionally-recorded Demo!

- Prediction Time
  - Edge: 2 - 2.5s /Image Vs. Cloud: 0.007 - 0.01s /Image.
  - Due to high prediction time, Edge can take up to 4-5s for detection in worst case.

- Image Stream lag
  - Edge: 0.009 - 0.02s vs. Cloud: 0.5 - 1s

- CMD lag
  - Edge: Negligible vs. Cloud: ~0.5s
Edge Cloud-Native Cluster

› Project Execution & Accomplishments
  › All 3 objectives completed with code and documentation committed.
  › The original Obj 3 of integrating the system with Clover is still open.
  › The Edge of Tomorrow project was accepted and delivered as a lightning talk at the Open Networking Summit, Europe, 2018!
Edge Cloud-Native Cluster

› Recommendations for future work
  › CI/CD pipeline for Edge/Edge-Cloud.

  › Better compatibility with Docker and kubernetes for ARM devices.

  › Low-cost GPU enabled Edge devices supporting popular deep learning frameworks.

  › Framework for designing collaborative ML algorithms for Edge-Cloud with ease.
Edge Cloud-Native Cluster

› Questions?