Edge of Tomorrow
Deploying Collaborative Machine Intelligence to the Edge

Adarsh Pal Singh
International Institute of Information Technology, Hyderabad

Wenjing Chu
Futurewei Technologies, Inc.
@wenjing
Do you like this “Edge of Tomorrow”? 

Well, let’s work on creating a better one…
AI Delivered through a Cloud Today

A camera
A Raspberry Pi
An actuator to control physical environment
A server
A storage service
A deep training engine
An inference engine + application
https://
So, What’s the Problem?

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

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

Local: <5-20ms
Local: OK MB/s
Improved privacy
Improved reliability, security
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
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
- A server
- A storage service
- A deep training engine
- A conventional app
- A model serving service
AI Delivered through an *Edge* Cloud of Tomorrow
AI Delivered through an *Edge* Cloud of Tomorrow

A camera
A Raspberry Pi
An actuator to control physical environment
A deep training engine
A storage service
A model serving service
A conventional app
A server
An inference engine
An image processor

Device
Edge
Data center

Local: <5-20ms
Local: OK MB/s
Improved privacy,
reliability, security
Less MB/s, save cost
OK 100’s ms latency
Let’s Experiment: Real Time Object Detection

YOLO: You Only Live–Look Once

Let’s Experiment: Latency Edge vs. Cloud

**Aim:** To trigger an action locally based on the detection of an object of interest.

**Demonstration 1**
- RPi Camera V2
- LED as a logical actuator
- Raspberry Pi 3 interfaced with sensor and actuator
- 720x480 image stream @ 30FPS
- Raw Data Stream over Internet
- Google Kubernetes Engine (Cloud)
- '0' or '1'

**Demonstration 2**
- RPi Camera V2
- LED as a logical actuator
- Raspberry Pi 3 interfaced with sensor and actuator
- 720x480 image stream @ 30FPS
- Raw Data Stream over Intranet
- Raspberry Pi Kubernetes Cluster (Edge)
- NN trained in cloud and off-loaded to edge
Some of the Details

- 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 does the following parallel tasks: (a) Video streaming and (b) Waiting for CMD from Edge/Cloud (for actuation).
- Pod workflow: Capture image stream -> Run NN -> Send CMD.
- Simple Socket programming used for sending/receiving CMD.
- Latency test: Edge vs. Cloud!
Demo: Let’s See it.

RPi3 on the edge vs. GKE with GPU on control loop latency running on YOLO object detection NN algorithm.

*YOLOv3: https://arxiv.org/abs/1804.02767
Some Numbers

- **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
Some Observations

- The *Edge of Tomorrow* is an intelligent one.
- The intelligent services can be realized with cloud methodology today, but have to address some crucial challenges
  - Latency requirement for real-time control
  - Bandwidth cost
  - Privacy concerns
  - Reliability and security concerns
- The cloud needs to be extended to the edge to address these problems.
- The edge node needs to have sufficient compute resources to fulfill these potentials.
Questions?

- Adarsh’s work is supported by an Linux Foundation Networking internship in the OPNFV Clover and Edge Cloud projects.
- Clover: https://wiki.opnfv.org/display/CLOV
- Edge cloud: https://wiki.opnfv.org/display/EC