DeePar: A Hybrid Device-Edge-Cloud Execution Framework for Mobile Deep Learning Applications

Yutao Huang[1], Feng Wang[2], **Fangxin Wang[1]**, Jiangchuan Liu[1]
[1]School of Computing Science, Simon Fraser University, Canada
[2]Department of Computer and Information Science, The University of Mississippi, USA

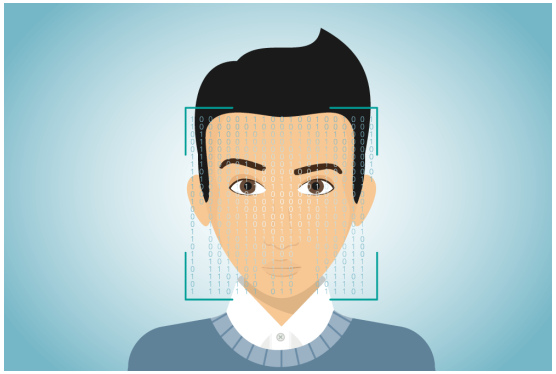
2019/04/29

Outline

- Introduction
  - – Today's challenges for mobile deep learning applications
- DeePar: Layer-level Partitioning Optimization for DNNs
  - – Enabling layer-level partitioning optimization for DNN inference
  - – Scheduling tasks for optimized total delay
  - – Experiment and simulation results
- Conclusion

Introduction

- AI is changing our lives



Face recognition

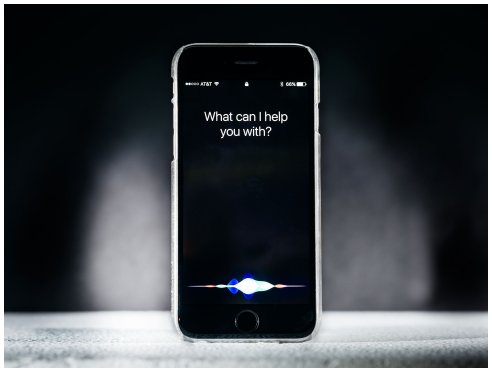
Machine translation

self-service supermarket

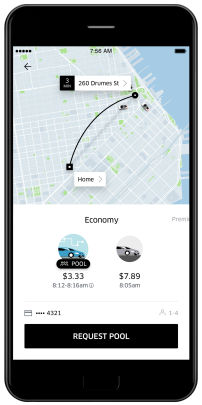
Driverless car

Introduction

- Rising in mobile deep learning applications

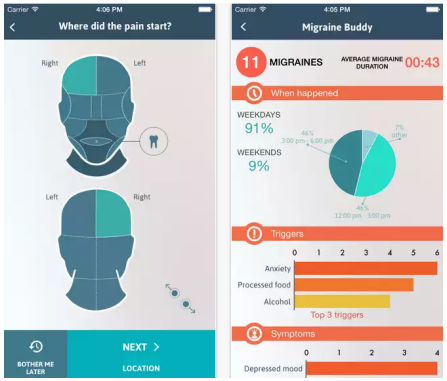


Apple Siri



Uber routing

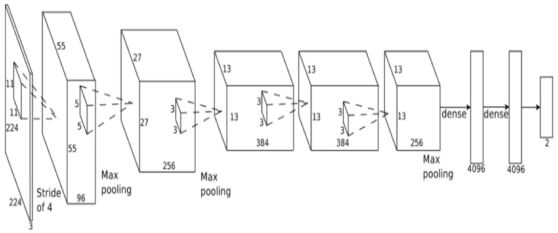
Face ID

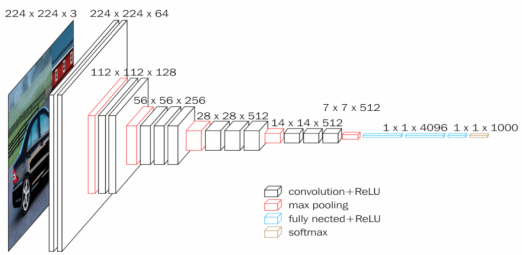


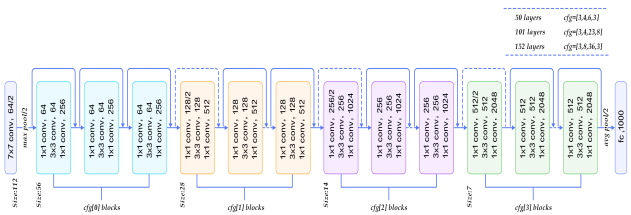
Migraine buddy

Introduction

- Models are getting larger:







8 layers
~16% error

AlexNet(2012)

19 layers
~7.5% error

VGG(2014)

152 layers
~3.5% error

ResNet(2015)

Picture downloaded from Internet

Introduction

- Cloud: current solution for mobile ML apps

Google Cloud NVIDIA GPU CLOUD

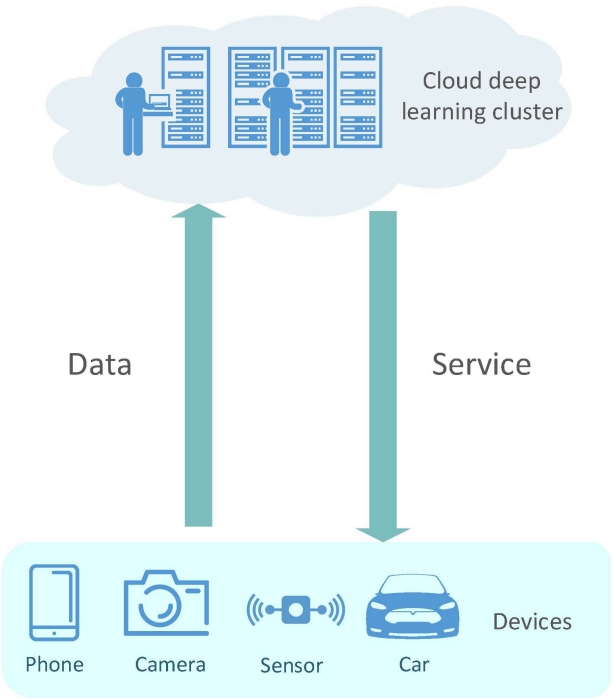
amazon web services™ Windows Azure 阿里云 aliyun.com

Amazon instances with GPU computing

GPU Instances - Current Generation		
p3.2xlarge	$3.06 per Hour	$0.27 per Hour
p3.8xlarge	$12.24 per Hour	$0.27 per Hour
p3.16xlarge	$24.48 per Hour	$0.27 per Hour
g3.4xlarge	$1.14 per Hour	$0.27 per Hour
g3s.xlarge	$0.75 per Hour	$0.188 per Hour

Introduction

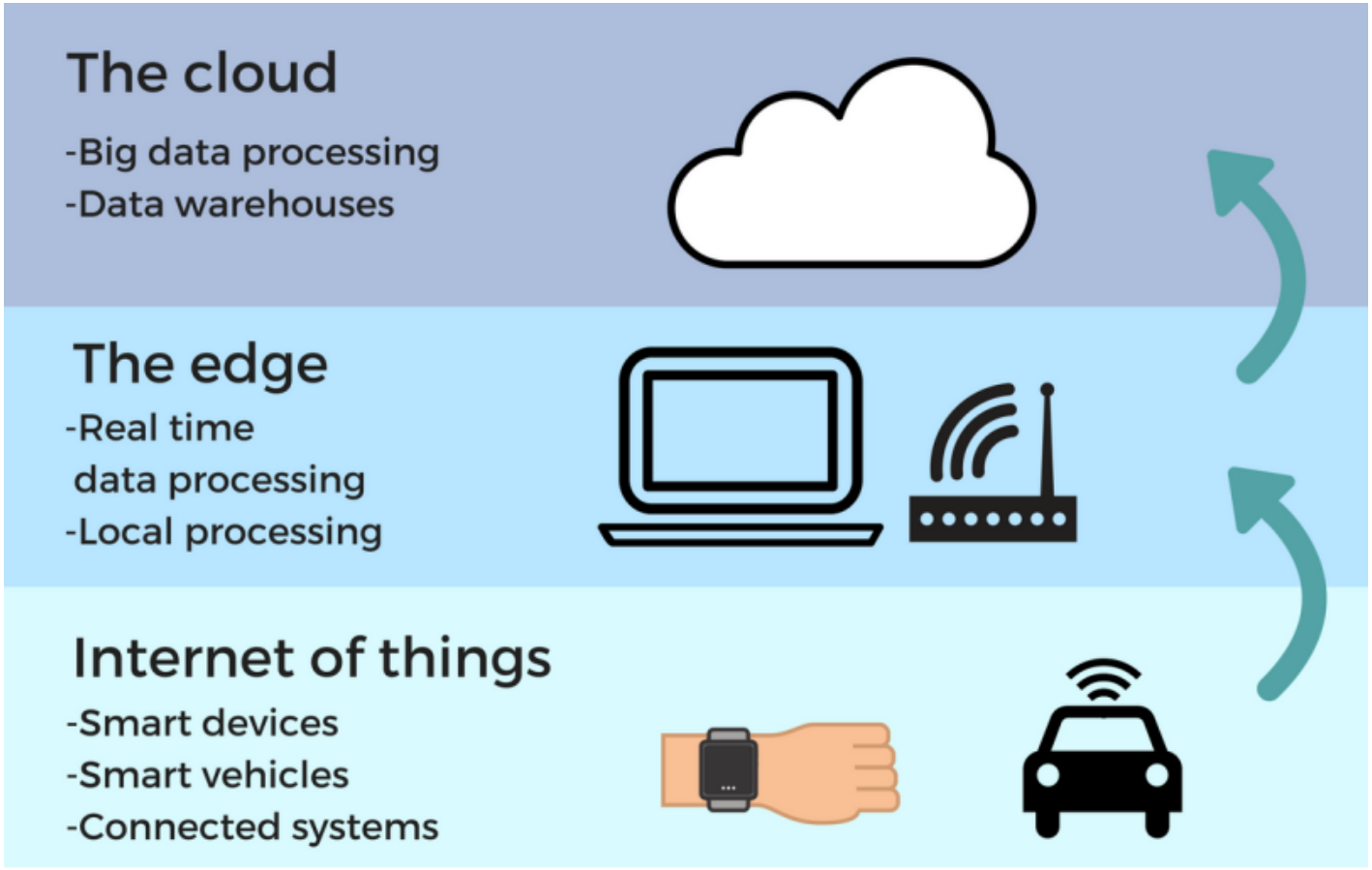
- Disadvantages of cloud computing



- Huge volume of Internet traffic
- Limited network bandwidth
- High latency response
- Security issue

Introduction

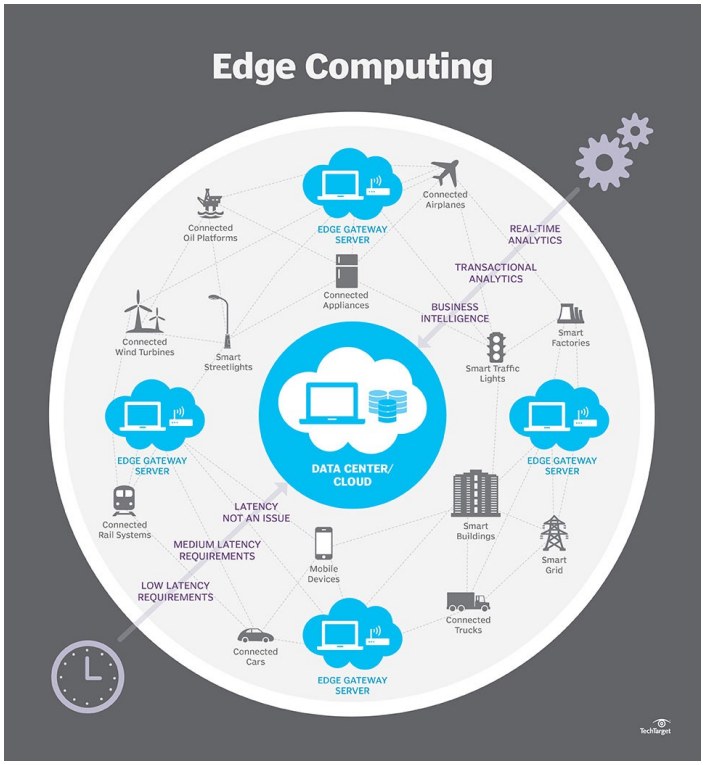
- Edge Computing:



Picture downloaded from Internet

Introduction

- Advantages of edge computing

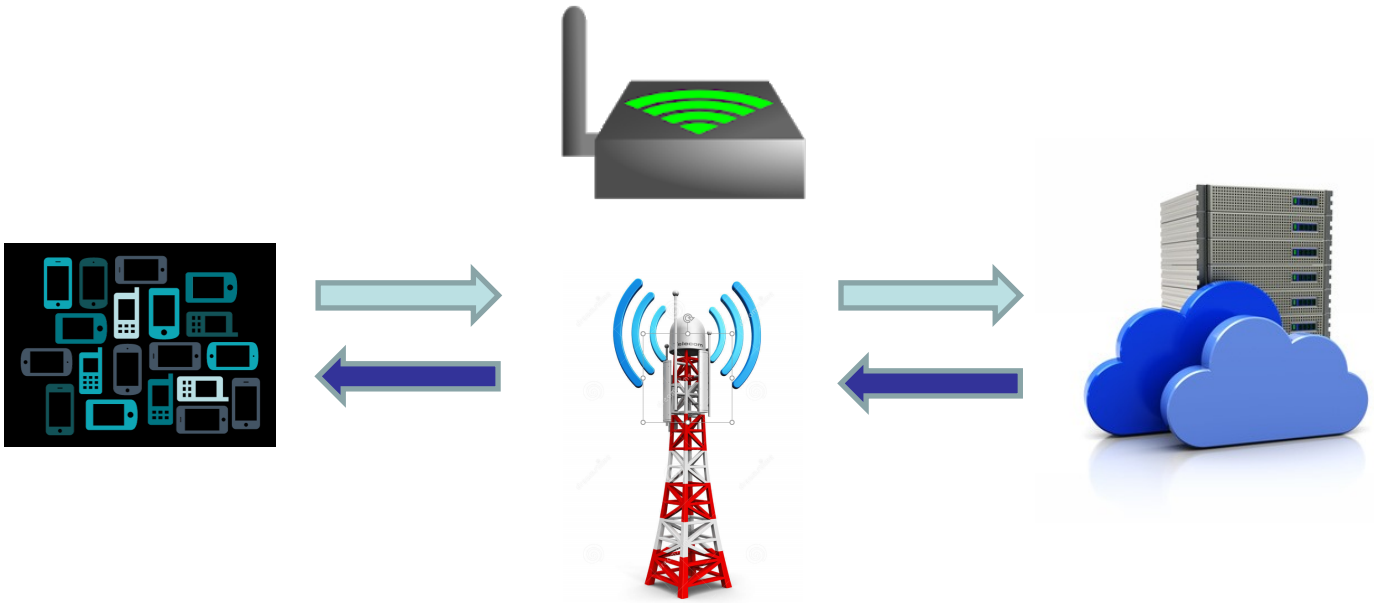


- Real-time or near real-time reaction
- Lower operating costs
- Reduced core network traffic
- Improved application performance

Picture downloaded from Internet

Introduction

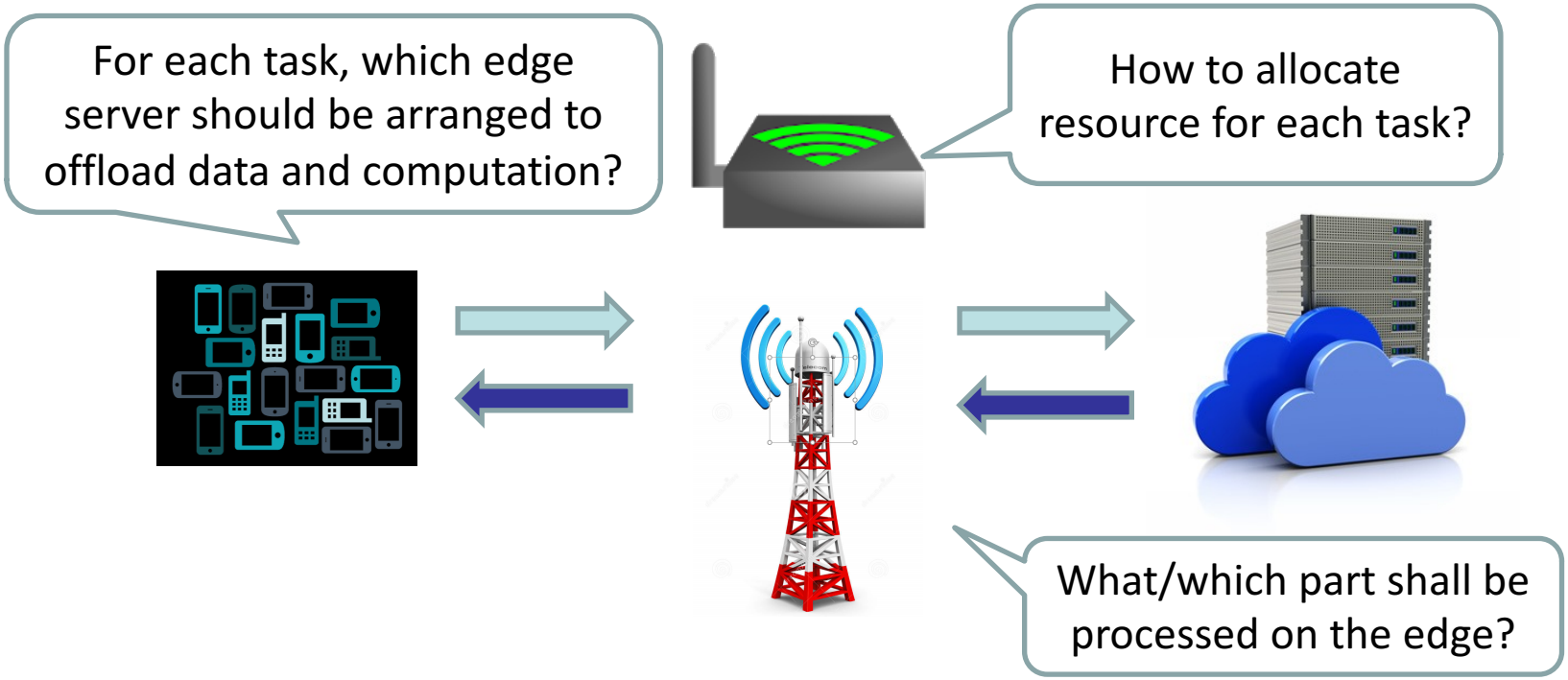
- Edge-assisted learning



Mobile device | Network edge | Remote cloud

Introduction

- Edge-assisted learning

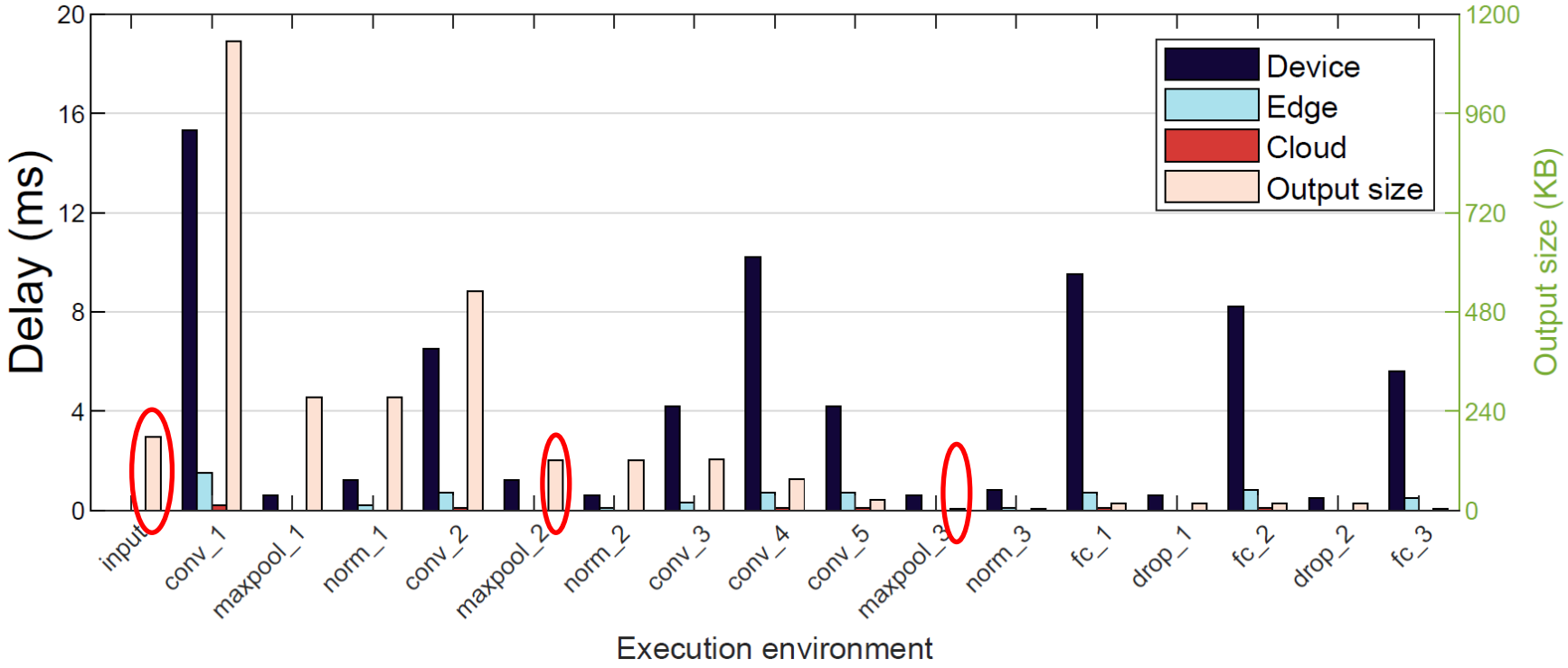


Outline

- Introduction
  - Today's challenges for mobile deep learning applications
- DeePar: Layer-level Partitioning Optimization for DNNs
  - Enabling layer-level partitioning optimization for DNN inference
  - Scheduling tasks for optimized total delay
  - Experiment and simulation results
- Conclusion

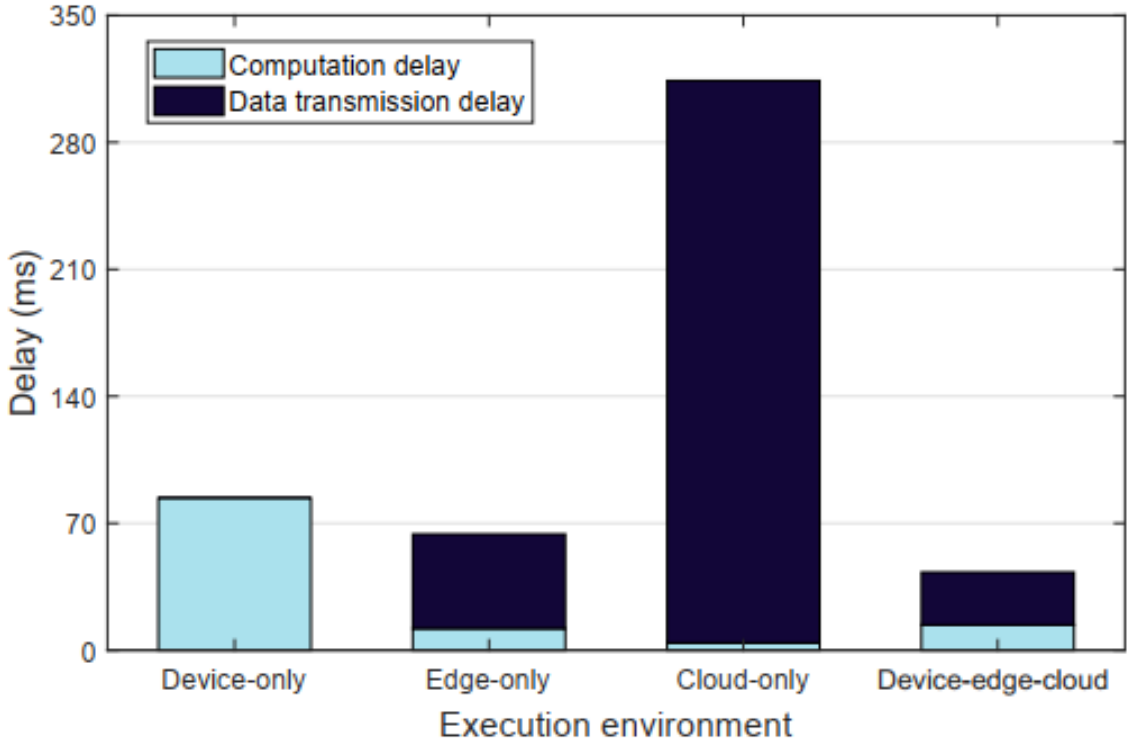
Motivation

- AlexNet layer-level performance



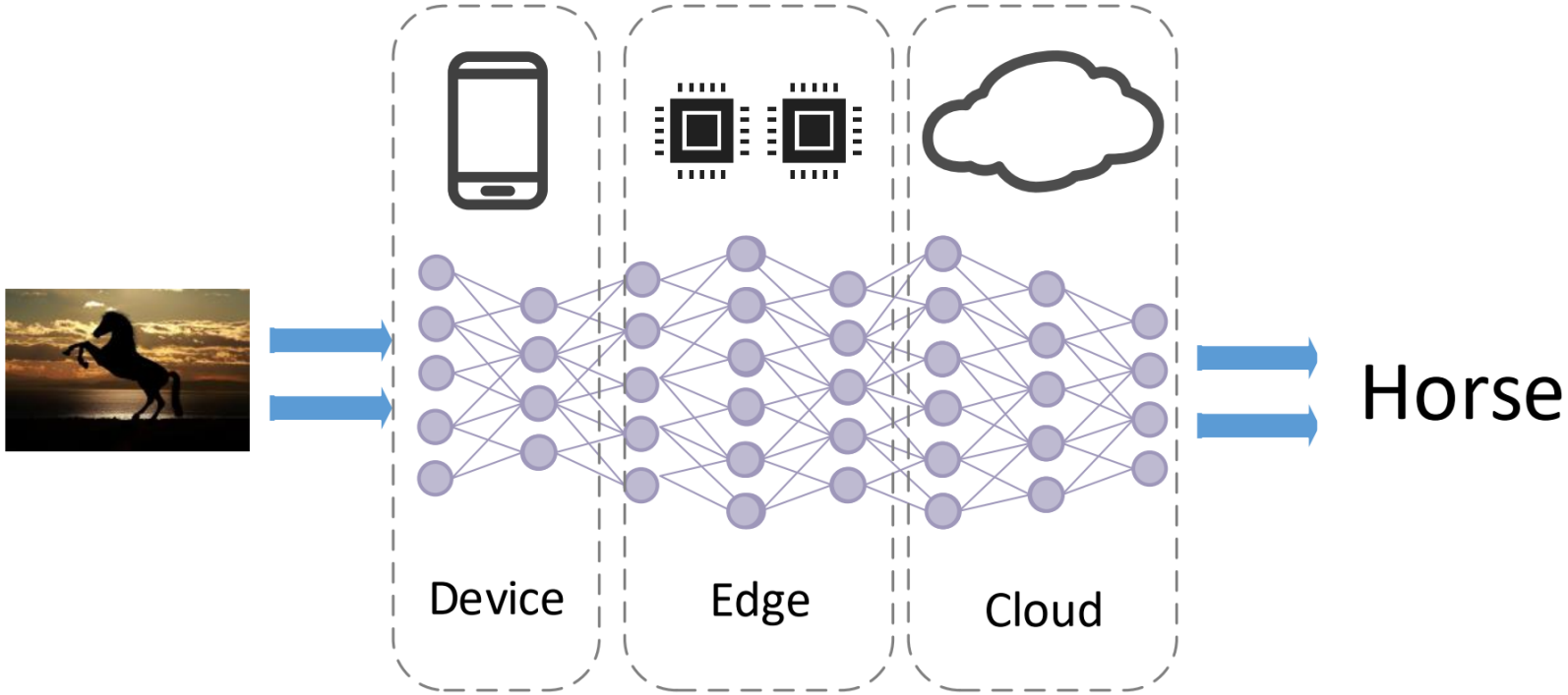
Motivation

- AlexNet performance comparison



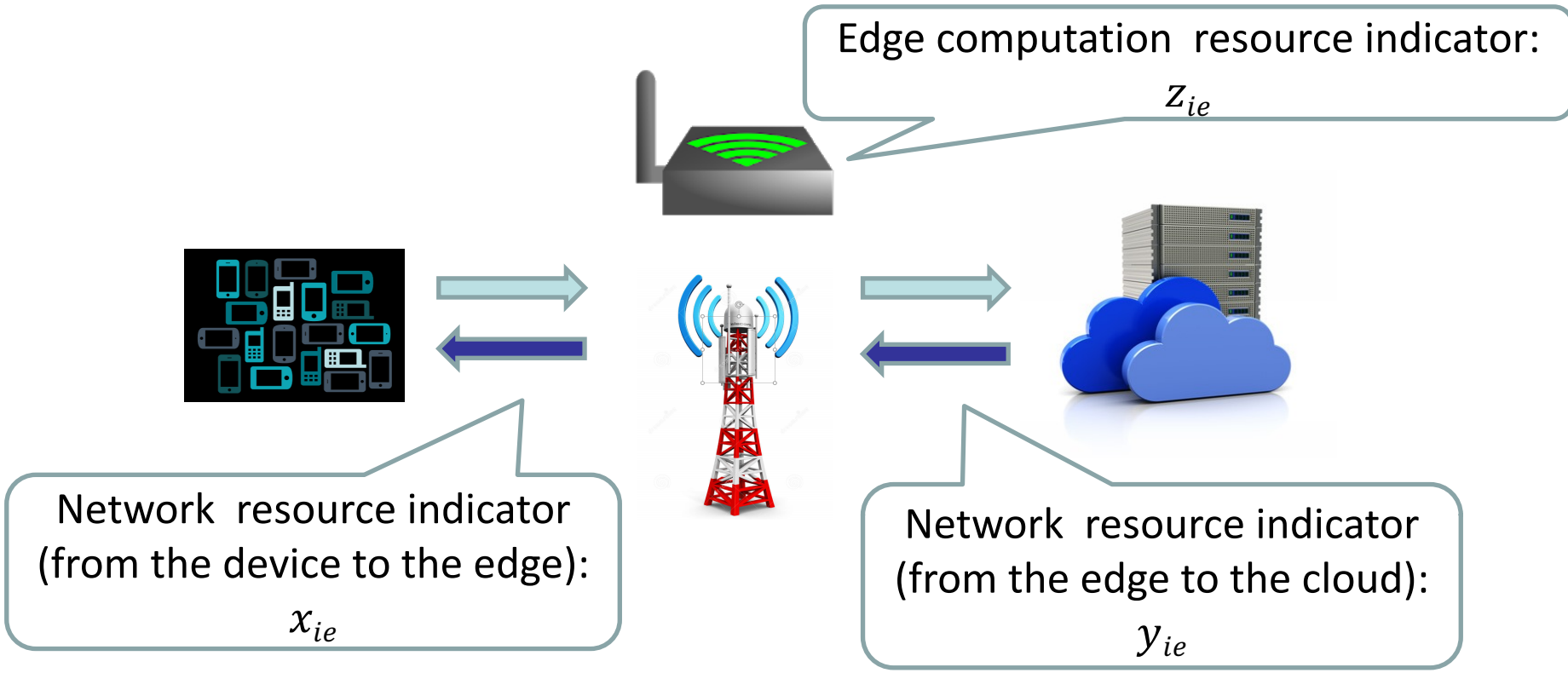
DeePar: A collaborative execution approach

- DeePar Framework for one single task



DeePar: A collaborative execution approach

- Online multi-task scheduling



Mobile device $i \in I$

Network edge $e \in E$

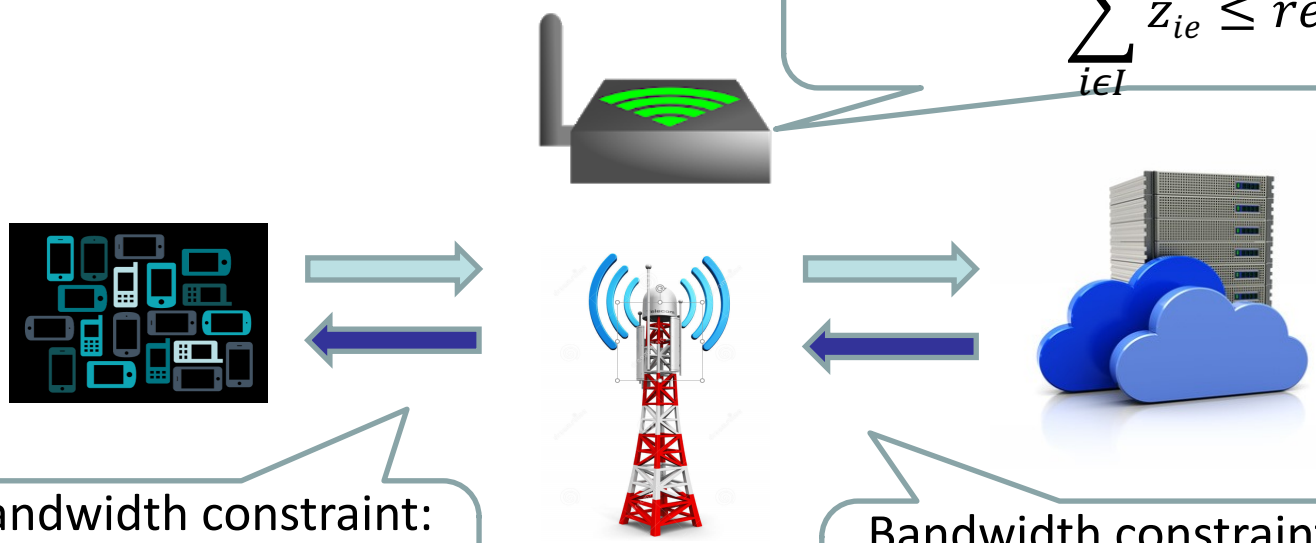
Remote cloud C

DeePar: A collaborative execution approach

- Constraints

Computation resource constraint:

$$\sum_{i \epsilon I} z_{ie} \leq re$$



Bandwidth constraint:

$$\sum_{i \epsilon I} b_i^1 * x_{ie} \leq B_e^1$$

Bandwidth constraint:

$$\sum_{i \epsilon I} b_i^2 * y_{ie} \leq Be^2$$

Mobile device $i \in I$

Network edge $e \in E$

Remote cloud C

DeePar: A collaborative execution approach

- Objective

Data transmission delay between device and edge server, and between edge server and cloud

Target: minimizing the total execution delay

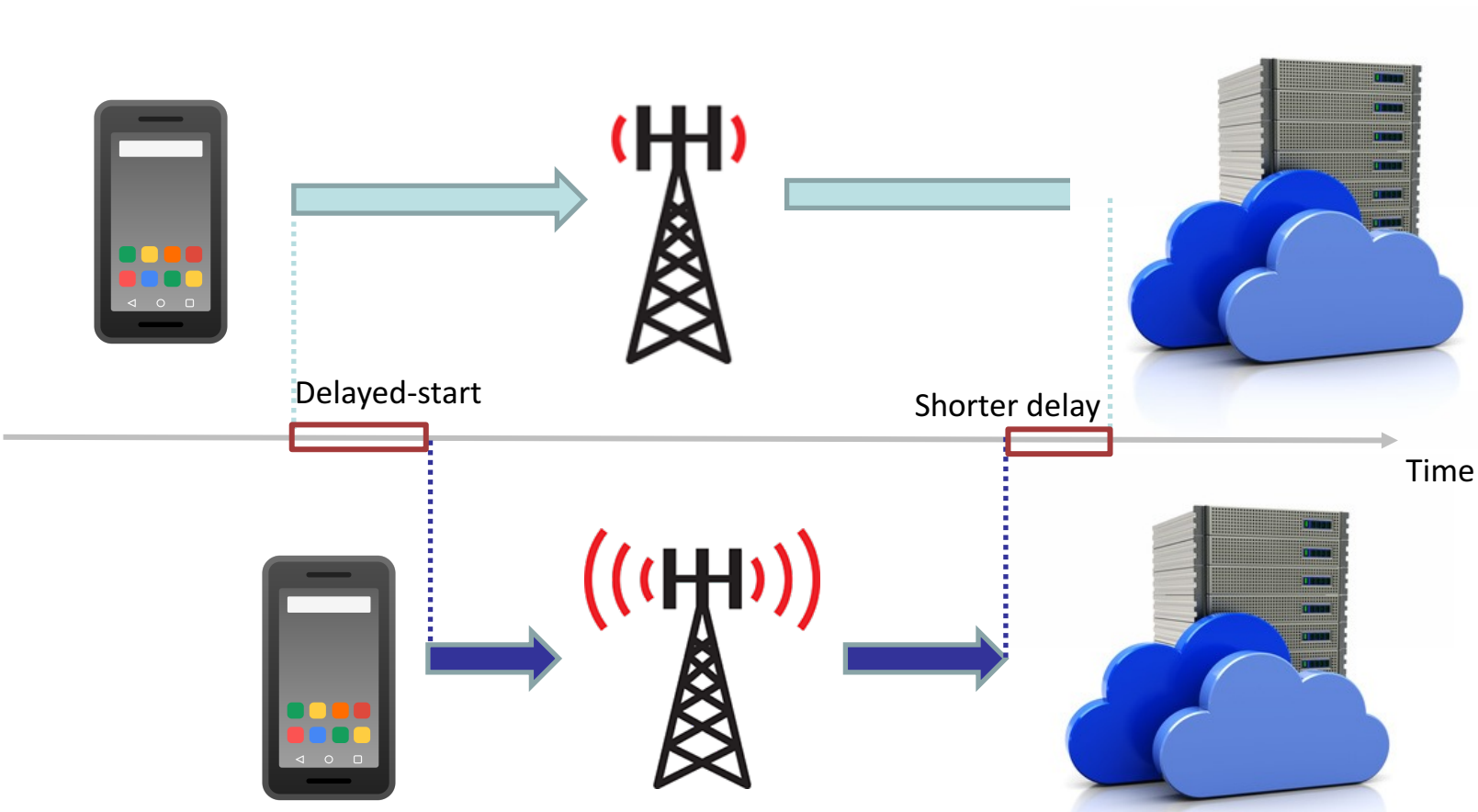
$$\min \sum_{i \in I} (\vec{t}_i^1 + \vec{t}_i^2 + \vec{t}_i^3 + \vec{t}_i^4 + \vec{t}_i^5 + \vec{t}_i^6)$$

Computation delay on device, edge server and cloud

Final result transmission delay (from cloud to the device)

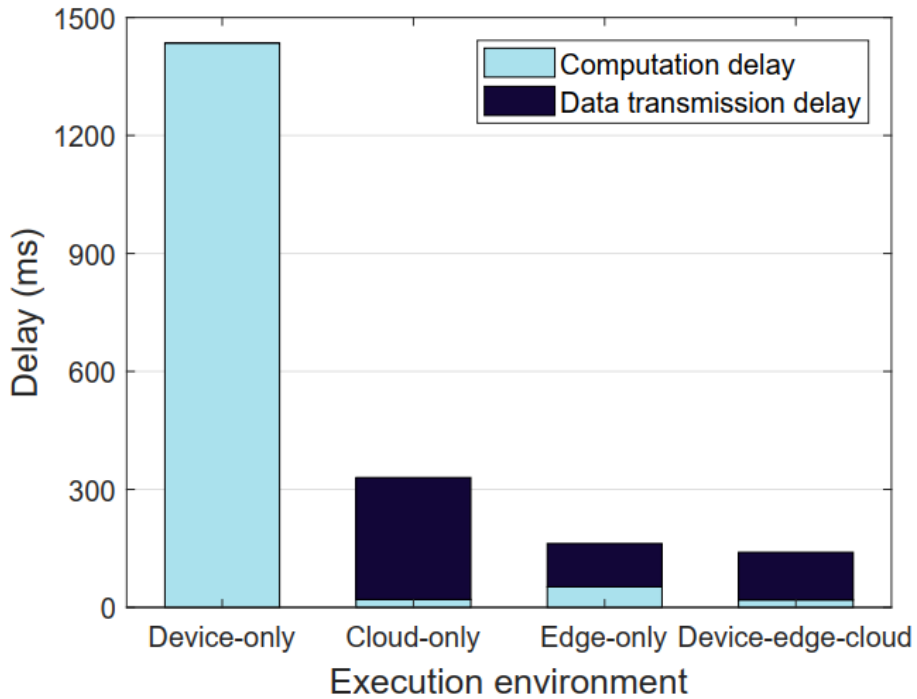
DeePar: A collaborative execution approach

- Delayed-start strategy



DeePar: A collaborative execution approach

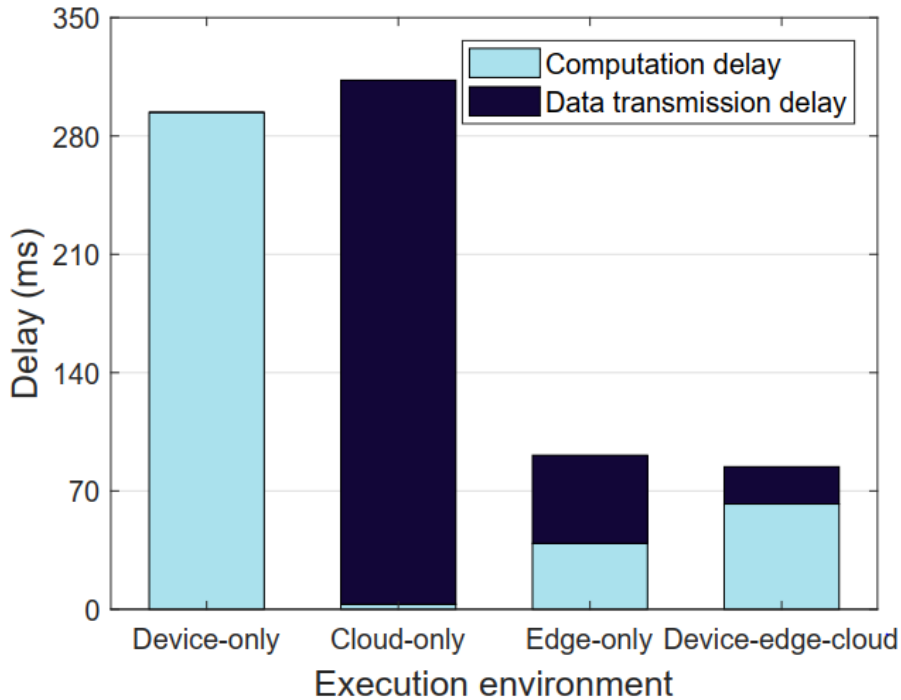
- Single-task experiment



DeepFace

DeePar: A collaborative execution approach

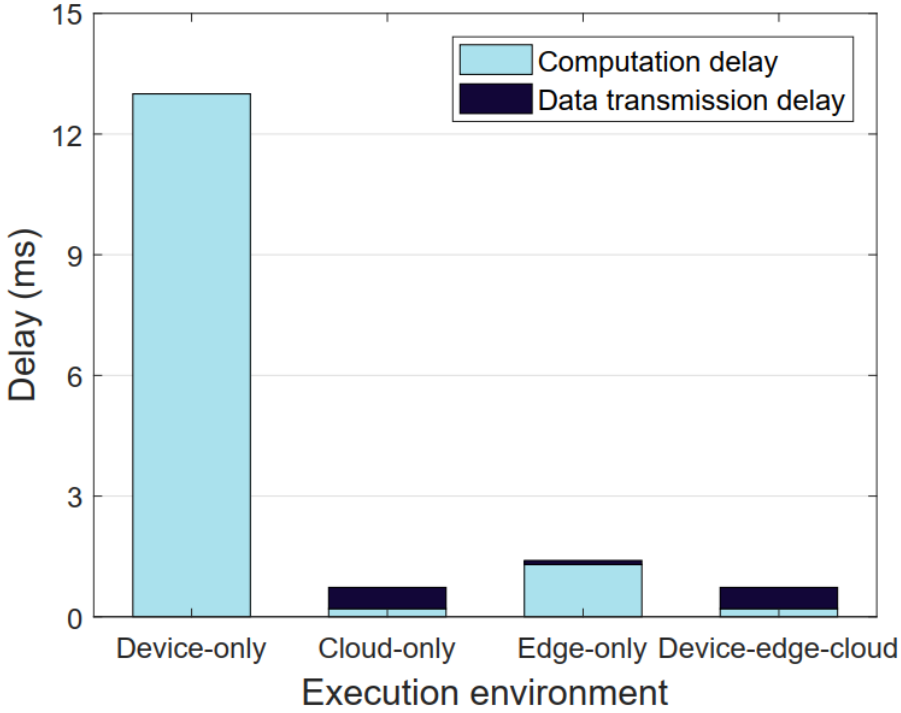
- Single-task experiment



VGG-16

DeePar: A collaborative execution approach

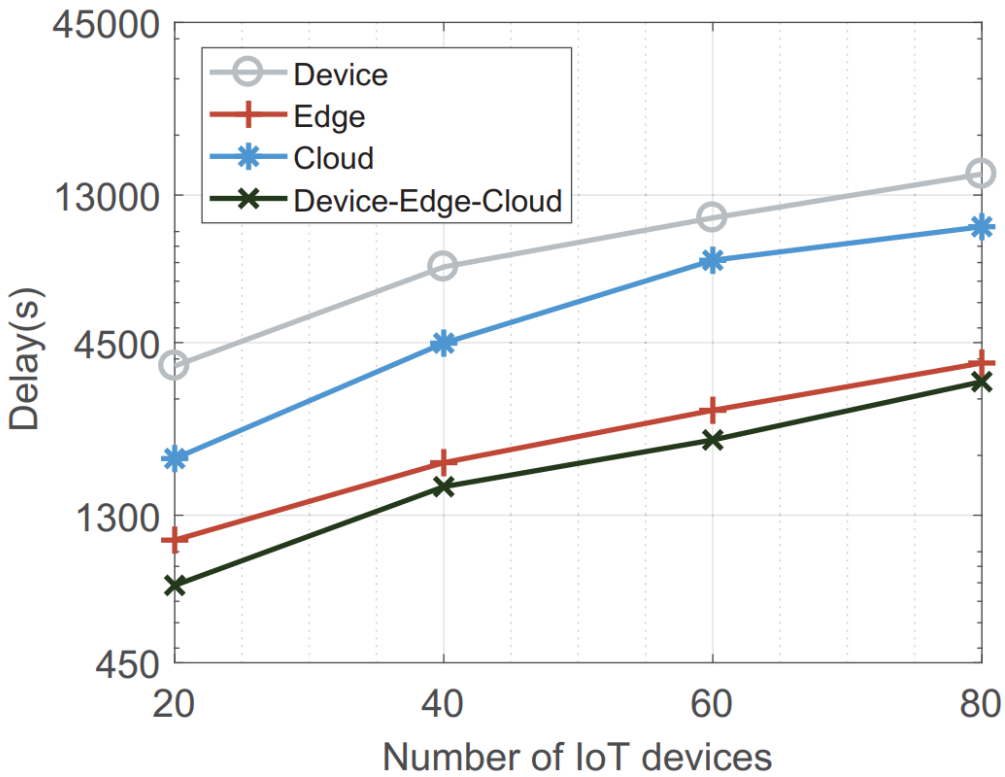
- Single-task experiment



LeNet

DeePar: A collaborative execution approach

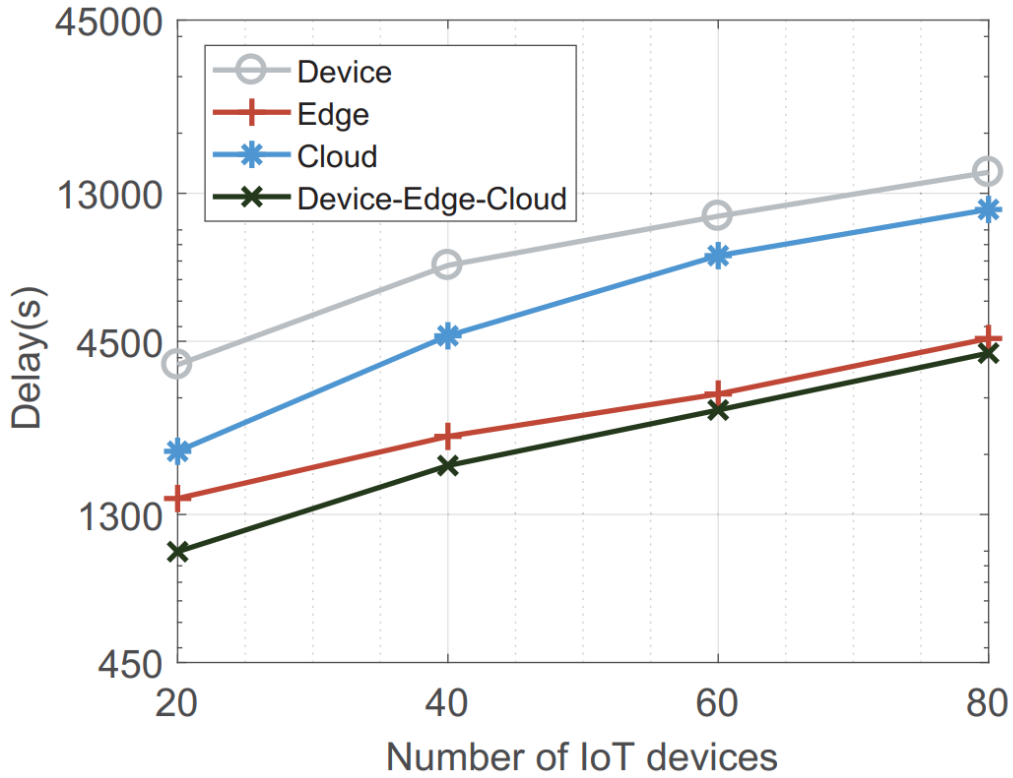
- Multi-task simulation



Time interval within 300s, 10 edge servers

DeePar: A collaborative execution approach

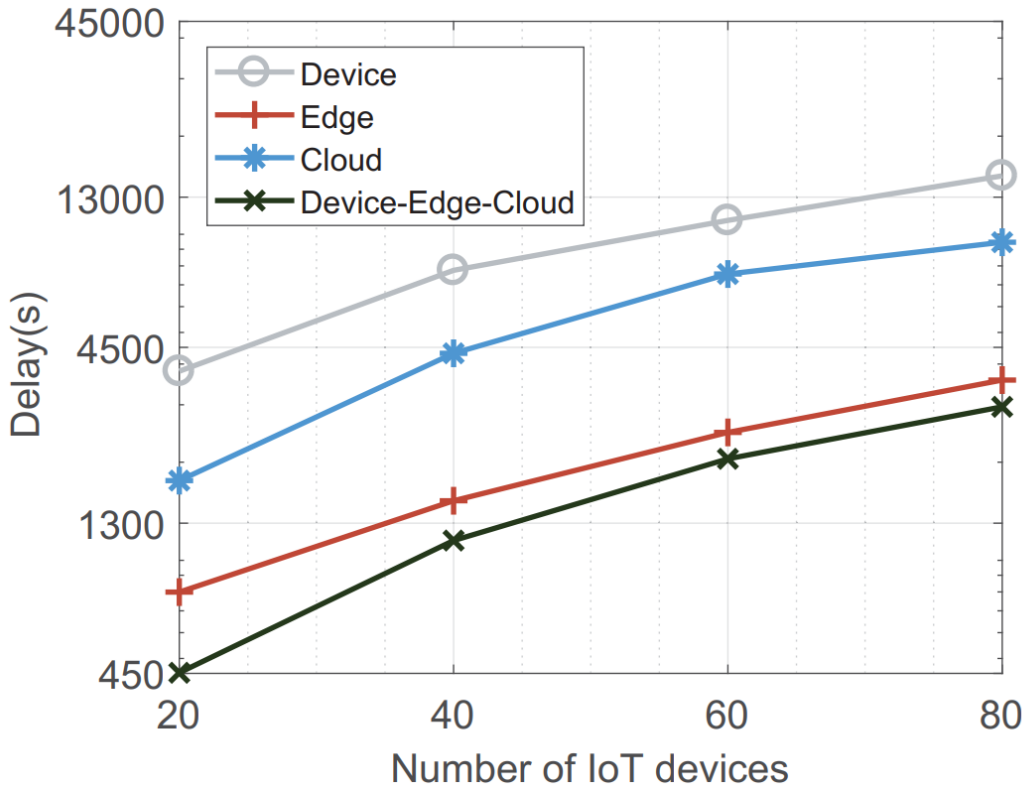
- Multi-task simulation



Time interval within 50s, 10 edge servers

DeePar: A collaborative execution approach

- Multi-task simulation



Time interval within 50s, 40 edge servers

Outline

- Introduction
  - – Today's challenges for mobile deep learning applications
- DeePar: Layer-level Partitioning Optimization for DNNs
  - – Enabling layer-level partitioning optimization for DNN inference
  - – Scheduling tasks for optimized total delay
  - – Experiment and simulation results
- Conclusion

Conclusion

- We propose DeePar, a **double-partition layer-level neural network partitioning** optimization framework for edge inference tasks.
- We formulate a **multi-task** scheduling problem for DeePar and propose an **online algorithm with a delayed-start** strategy.
- Through experiments and simulations, DeePar can outperform device-only, edge-only and cloud-only execution with **20% - 80%** delay reduction.