Distributed Analytics in Fog Computing Platforms Using TensorFlow and Kubernetes

Pei-Hsuan Tasi, Hua-Jun Hong, An-Chieh Cheng, and Cheng-Hsin Hsu

Department of Computer Science, National Tsing Hua University, Taiwan

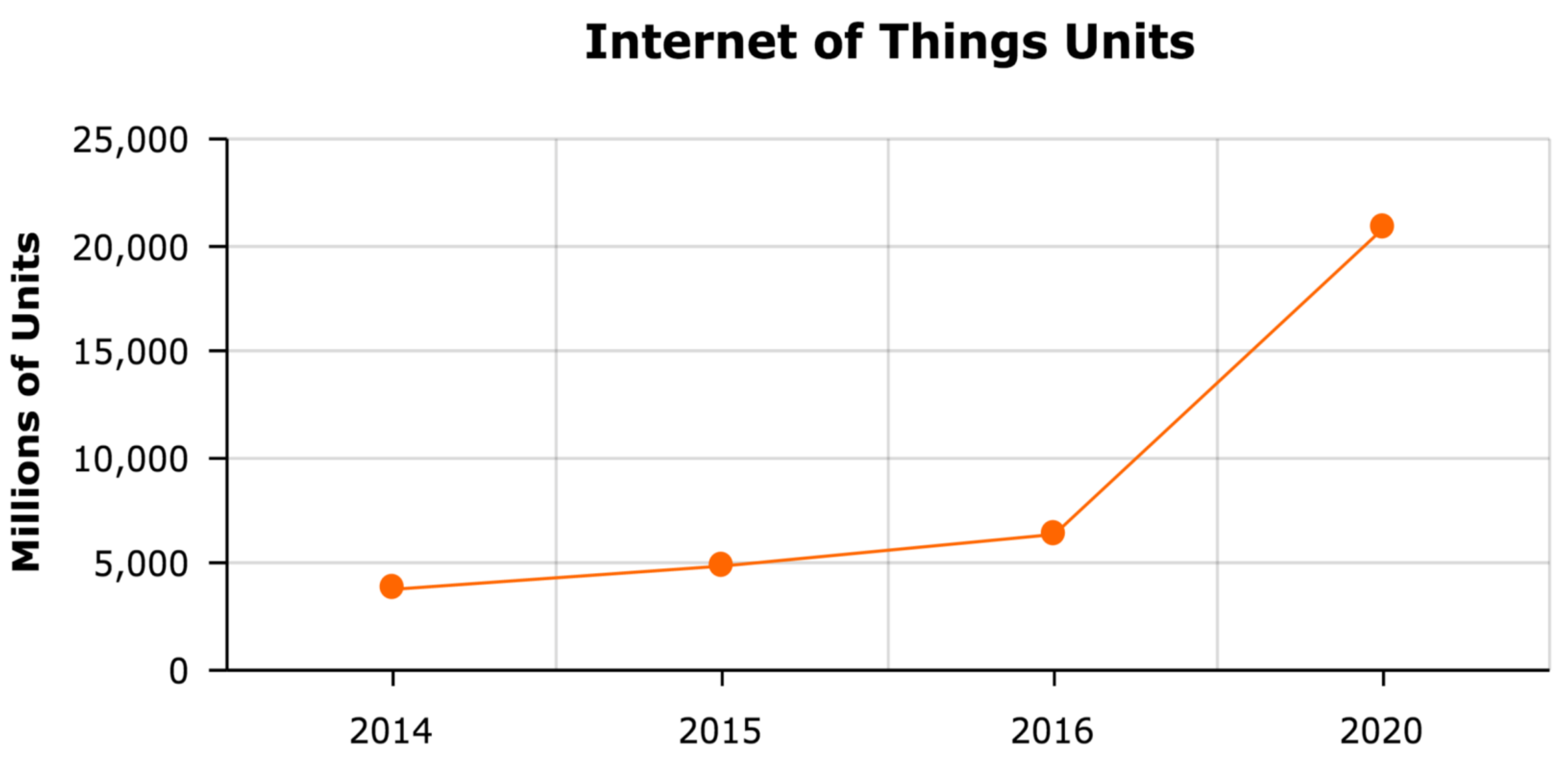
What is Fog?

- "Fog is a cloud close to the ground"
  - Bonomi et al. (2012) [1]
- Designed for Internet of Things (IoT)
  - Location awareness
  - Realtimeness
  - Mobility support
  - …

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli,"Fog computing and its role in the internet of things,"in *Proc. of ACM Workshop on Mobile Cloud Computing (MCC)*, 2012.

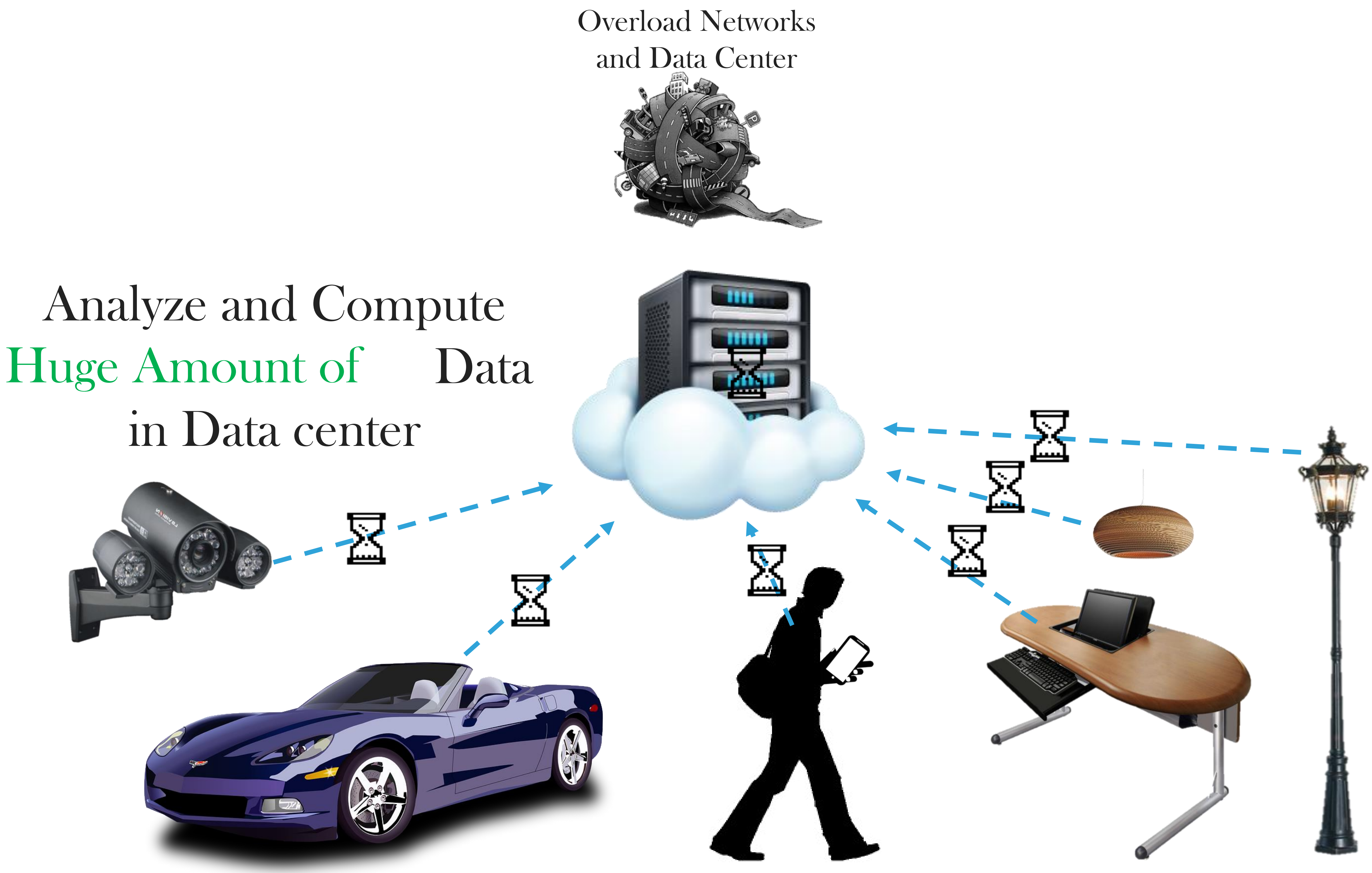
IoT Market is Soared!

- Internet of Things (IoT) grows rapidly
  - Devices: 20.4 billion in 2020 [1]
  - Market: $1.9 trillion in 2013 to $7.1 trillion in 2020 [2]
- Produce incredible amount of data
  - Overload the data centers and congest the networks seriously

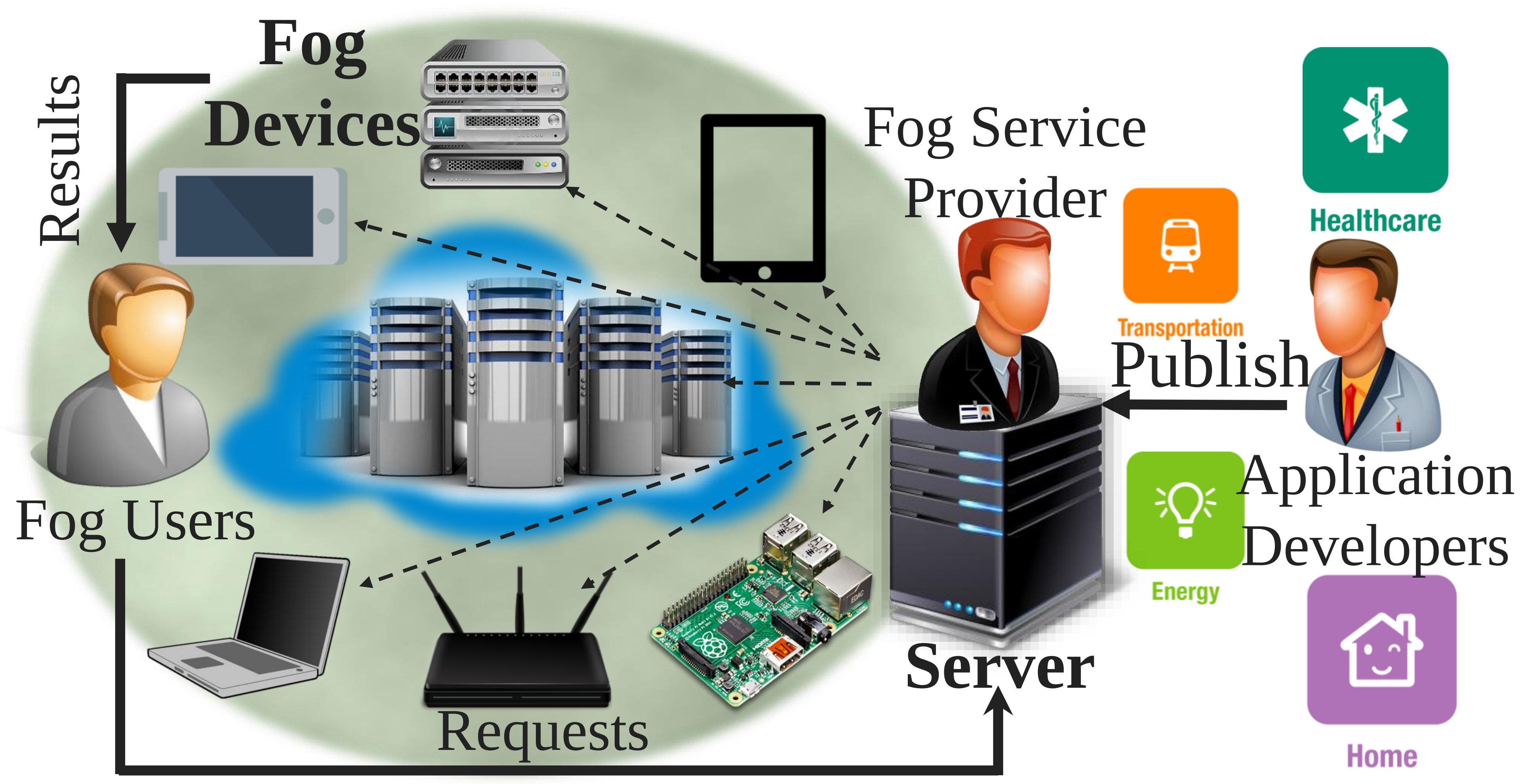


[1] http://www.gartner.com/newsroom/id/3598917
[2] http://www.ironpaper.com/webintel/articles/internet-of-things-market-statistics/

Limitations of Current (Cloud) Solution

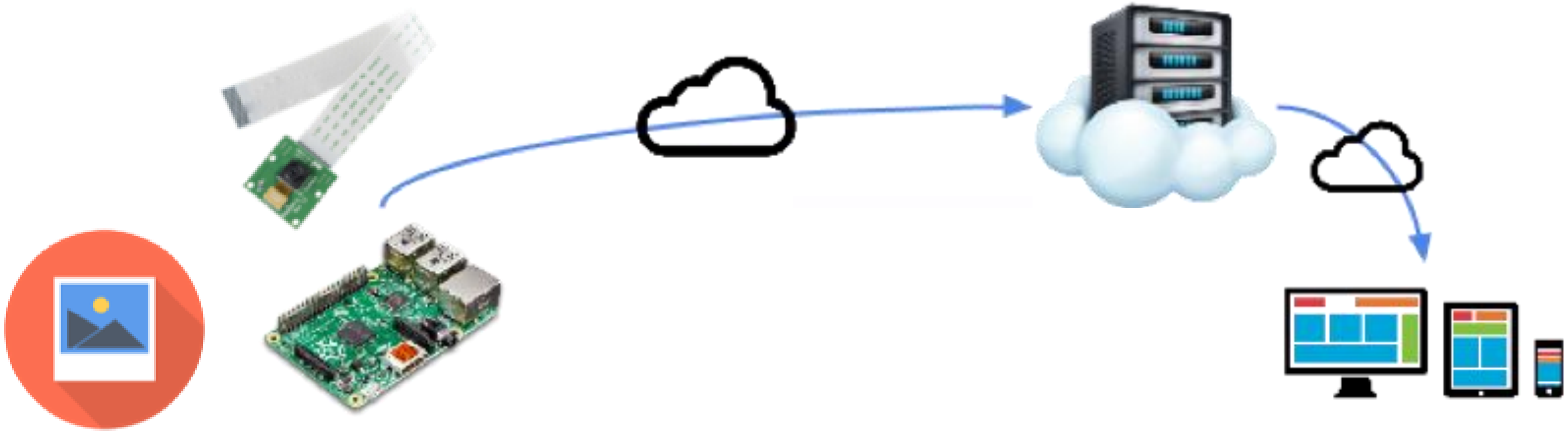


Fog Computing Ecosystem



Benefits of Fog Computing

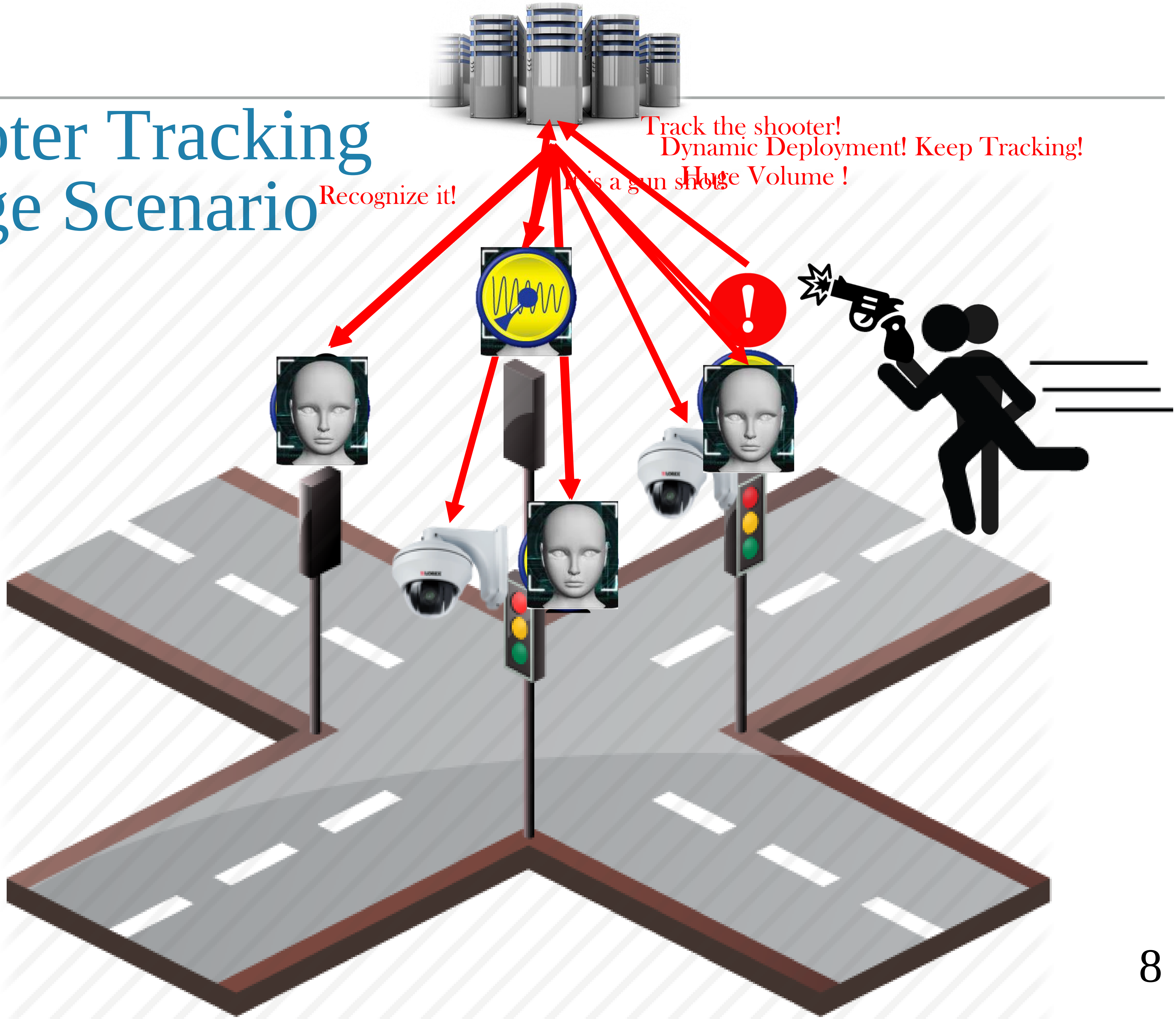
- Reduce latency
- Reduce network traffic
- Reduce the load of data centers
- Reduce privacy issues



Challenges

- Limited resources of the fog devices
  - How to run complex analytics?
- Mobility-required IoT applications
  - How to transparently and dynamically deploy applications?
- Many fog devices and IoT applications
  - How to efficiently manage them?

Shooter Tracking
Usage Scenario



Our Goals

- Dynamic deployment
- Efficient management
- Real-time analytics

Implement the Fog Computing Platform

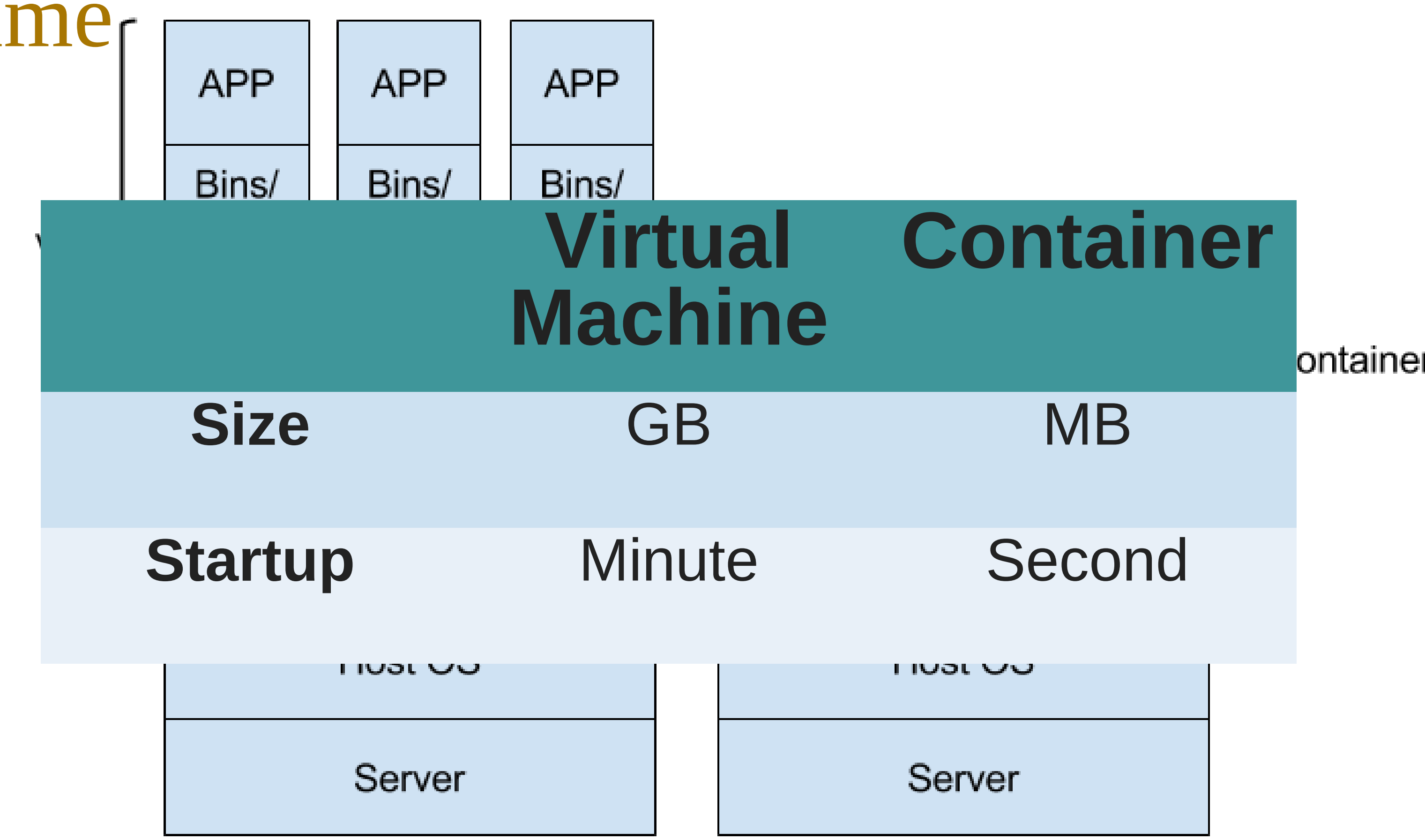
- Enhance open source projects
  - Docker -> Dynamic deployment
  - Kubernetes -> Management
  - TensorFlow -> Real-time analytics

Docker: Dynamic Deployment

- Container
  - LXC, Docker, …
- Virtualize analytics applications
  - Dynamic deployment
  - Easy migration
  - Efficient management
- Virtualization overhead
  - We implement a online regression mechanism to derive system models

Traditional VM v.s Container

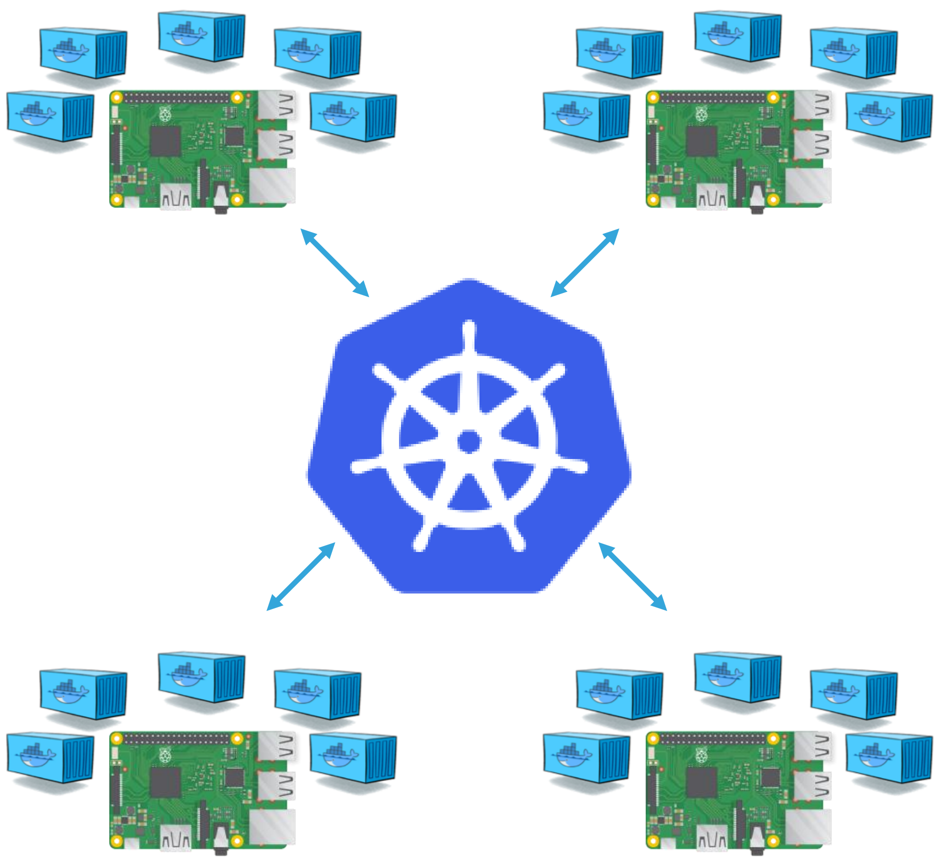
- Traditional Virtual Machine
  - Better isolation
- Container: light-weight VM
  - Smaller storage space, less computing power, and shorter startup time



	Virtual Machine	Container
Size	GB	MB
Startup	Minute	Second

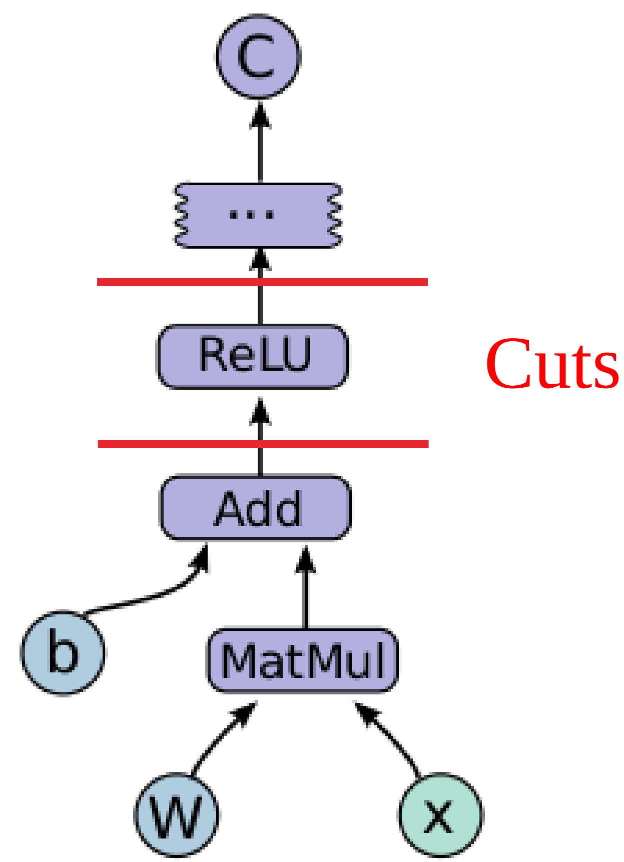
Kubernetes: Management

- Monitoring
  - Resource usage (CPU, Memory)
  - Containers status
  - Locations
  - Sensors
- Deployment
  - Deployment algorithm



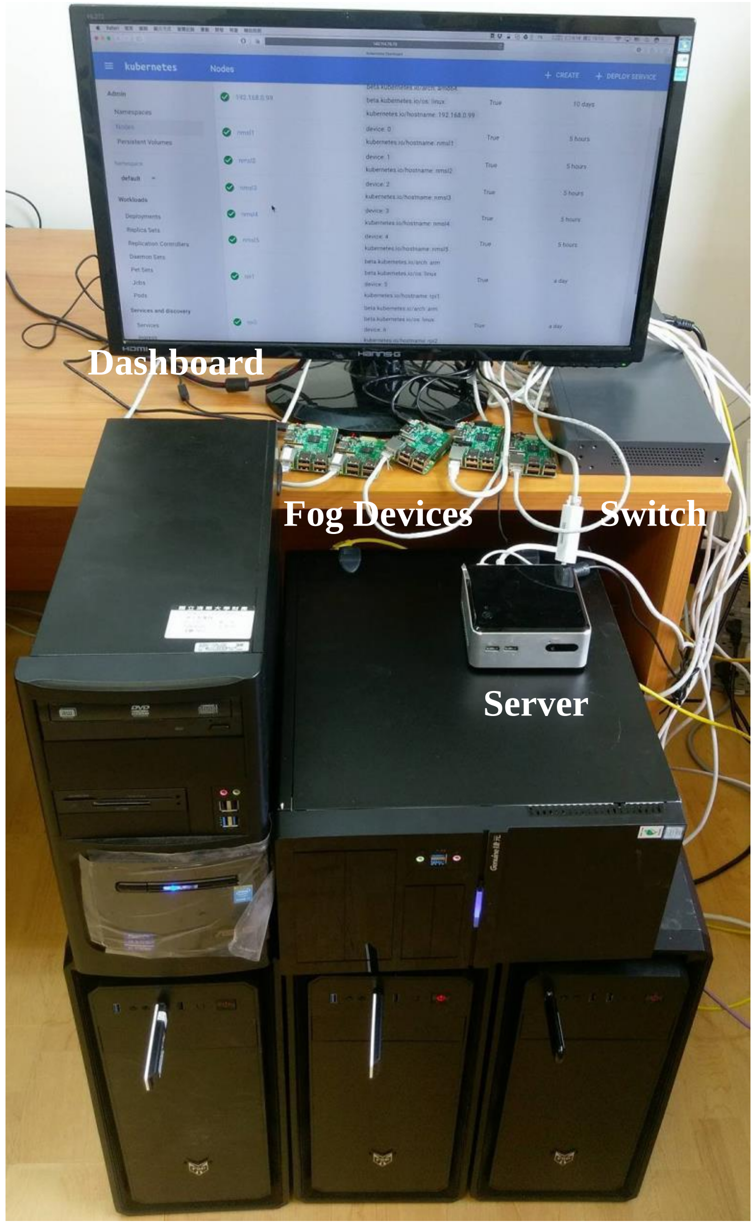
TensorFlow: Real Time Analytics

- An open-source programming model for Machine Intelligence
- Data flow graphs
- Distributed computing supported
  - We add queues to speedup distributed computing



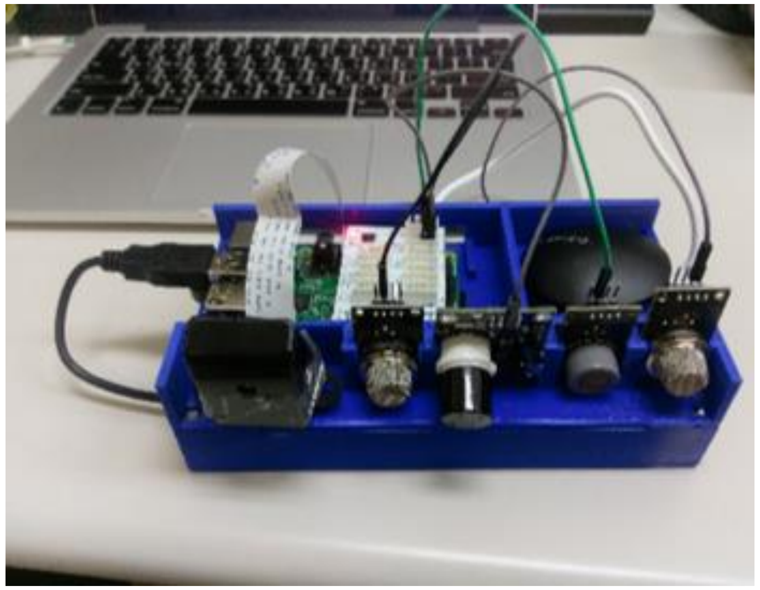
Testbed

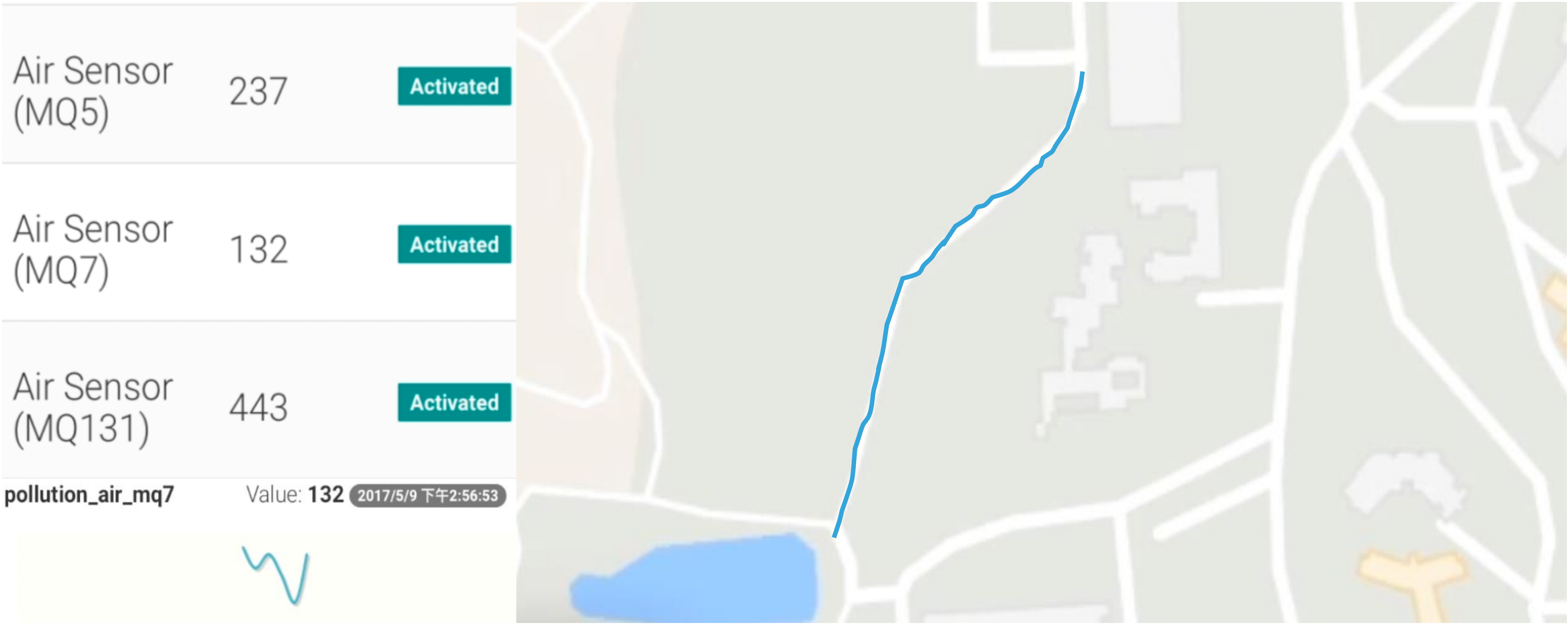
- Server
  - I5 CPU PC with Kubernetes
- Fog Devices
  - 5 Raspberry Pi 3 (1.2 GHz 4-core CPUs)
  - Kubernetes and Docker (TensorFlow is installed in Docker containers)



Application: Air Pollution Monitor

- Air pollution sensors installed on Raspberry Pi
- Moving average



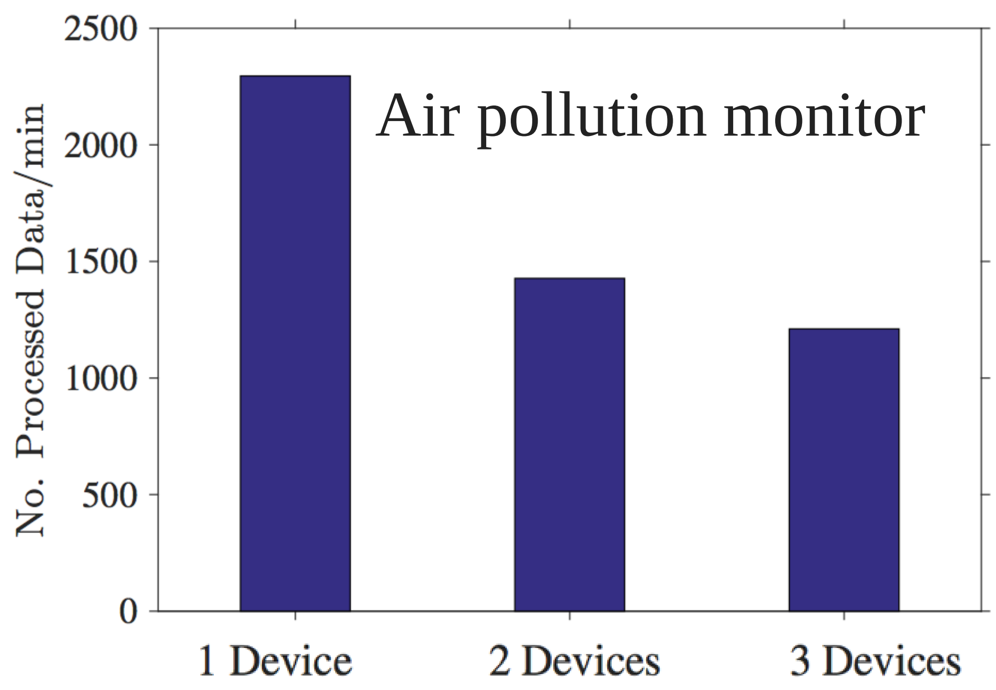


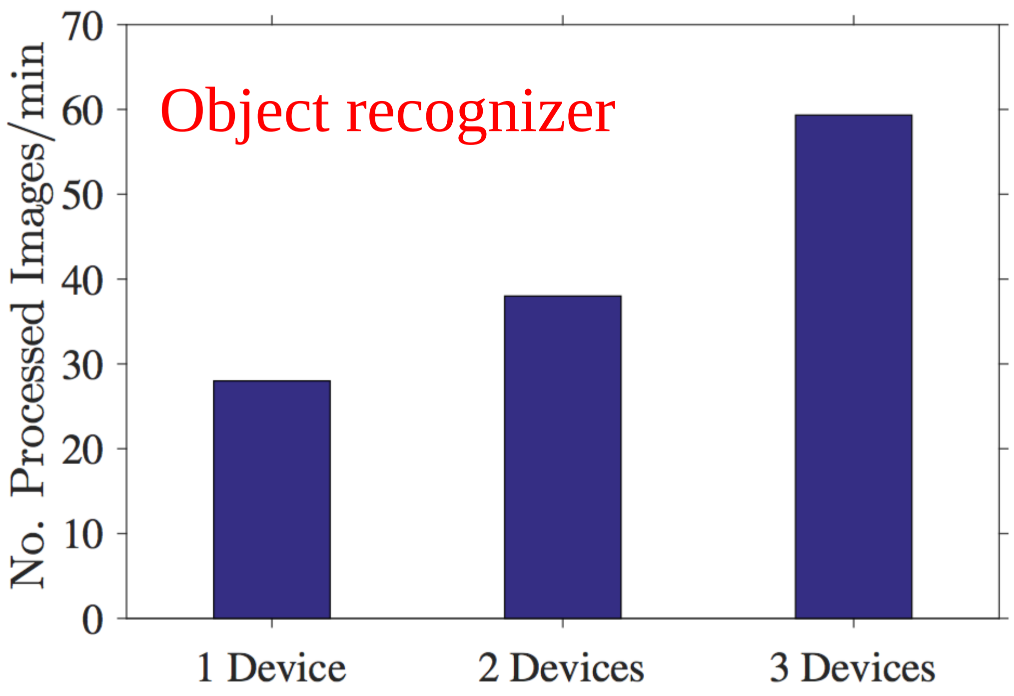
Application: Object Recognizer

- A pre-trained model (YOLO)
- Implemented using neural networks

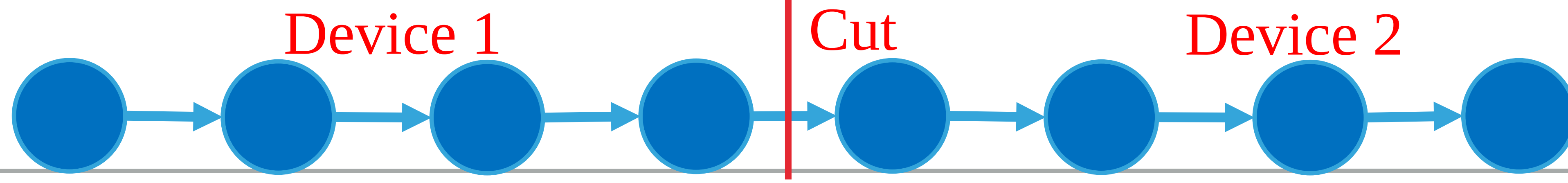
Distributed Analytics Result in Better Performance for Complex Analytics

- Air pollution monitor: too simple to benefit from distributed analytics
- Object recognizer:
  - One more device gives 35.5% and 54.1% improvements



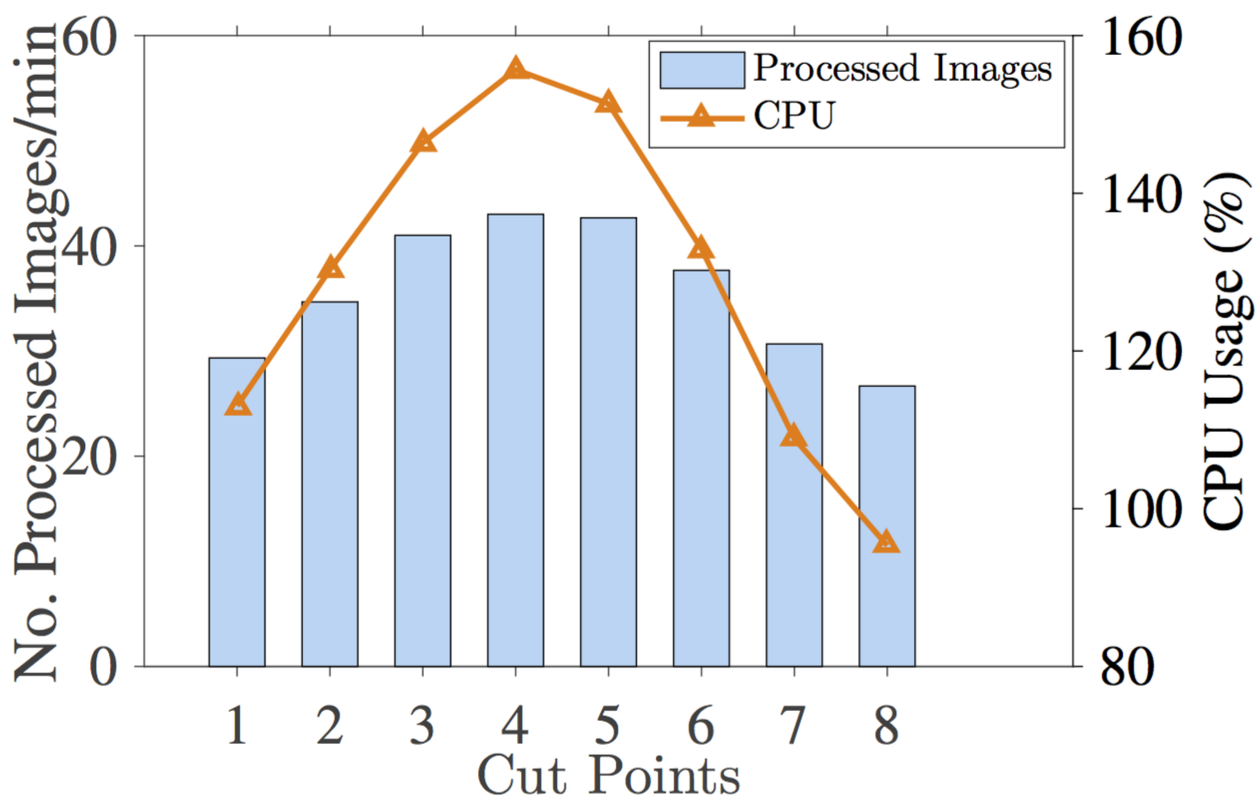


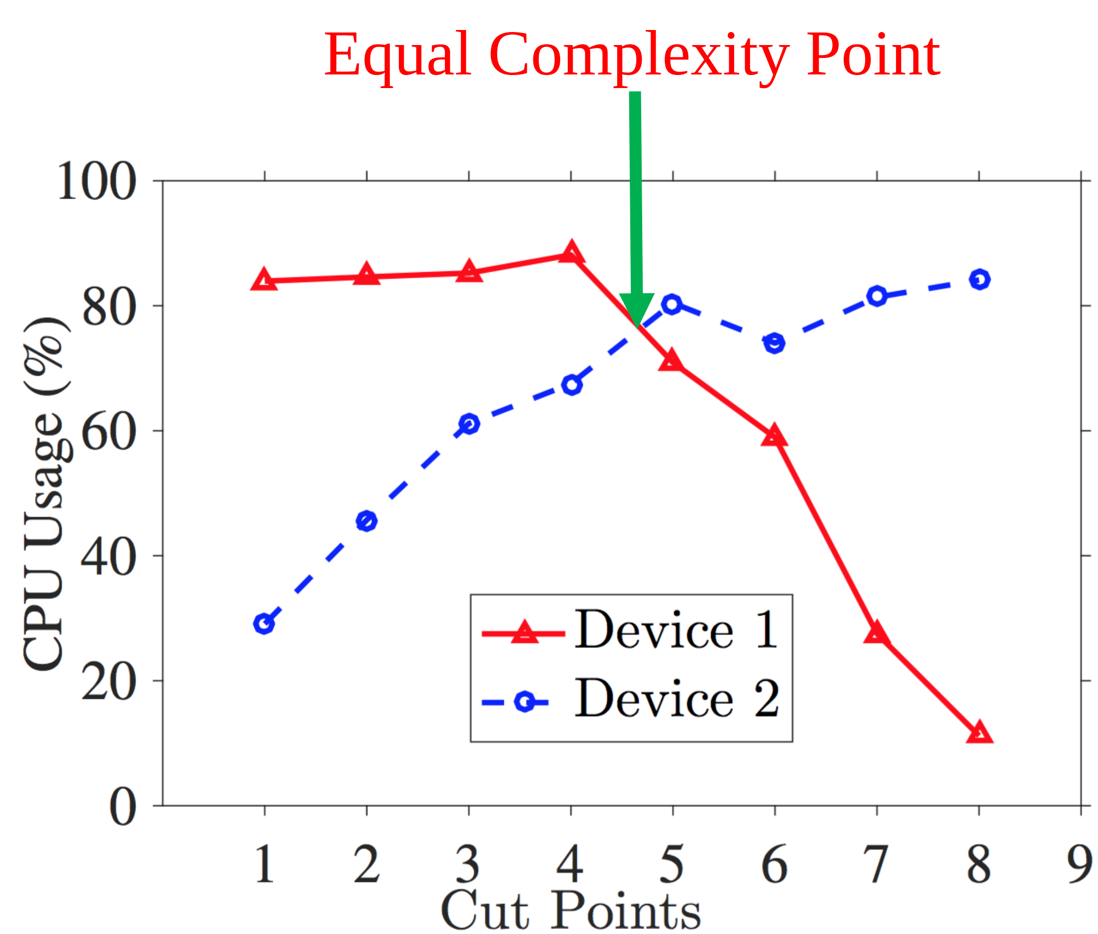
We only consider the object recognizer in the following results

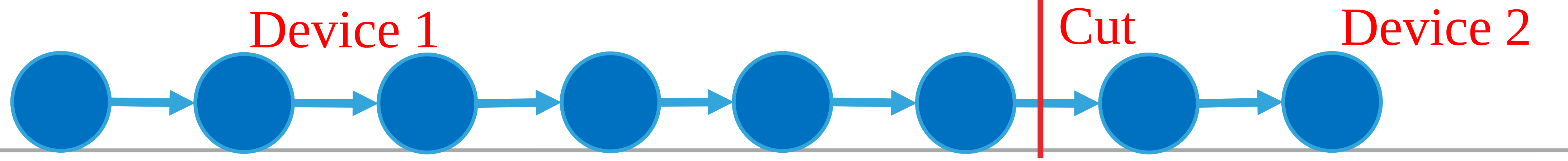


Cut on Equal Complexity Point Results in Better Performance

- Setup: run object recognizer on two fog devices (different cut points)
- Cut point 4&5 result in the most no. processed images
- Two fog devices consume the same CPU resources on cut point 4&5

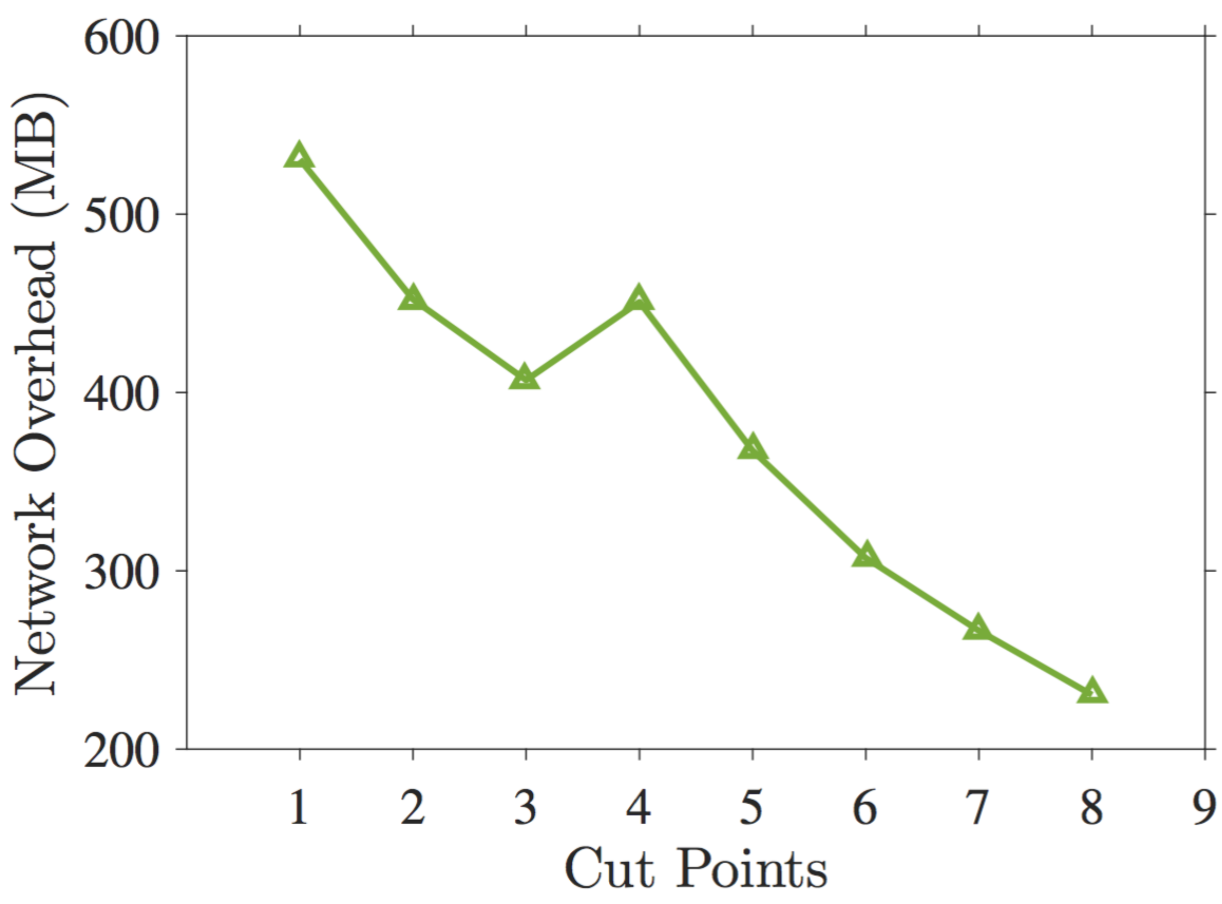






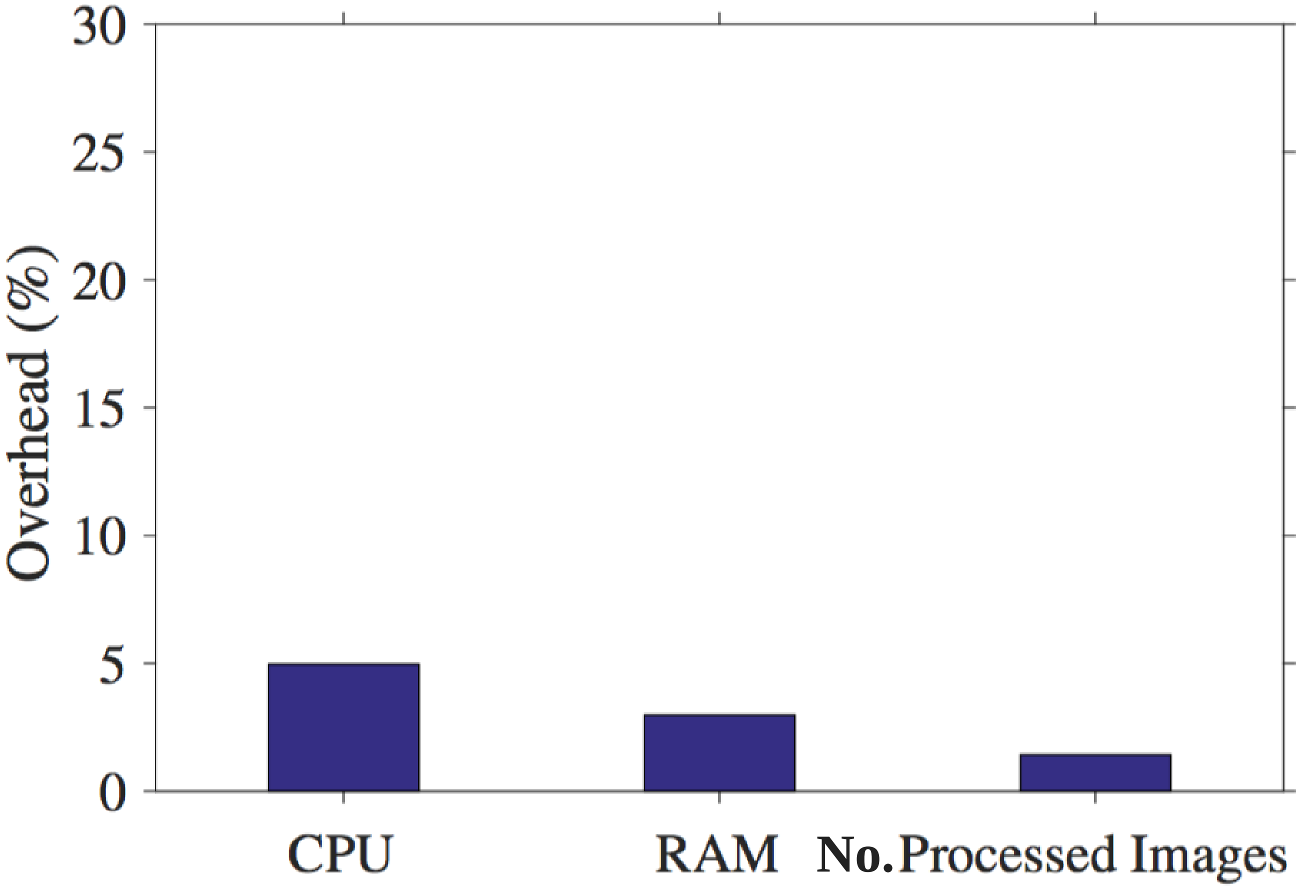
Cut Points Affect Network Overhead

- Put more computations on device 1 can reduce network overhead



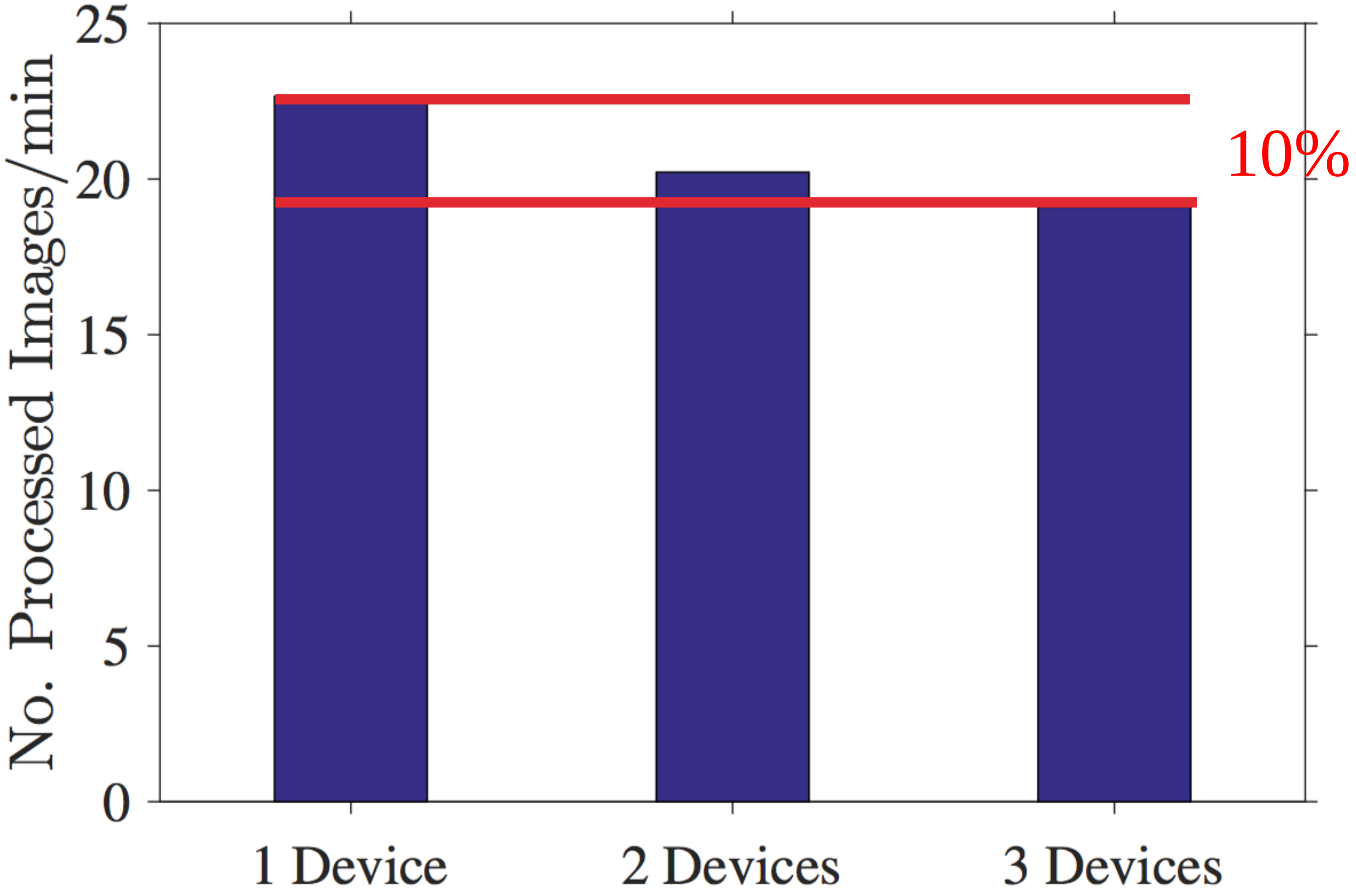
Low Virtualization Overhead

- Setup: with and without Docker
- Overhead from Docker
  - Less than 5%



Low Communication Overhead

- Setup: without Docker and distributed speedup
- Overhead from distributed computing using TensorFlow
  - 10%



Summary

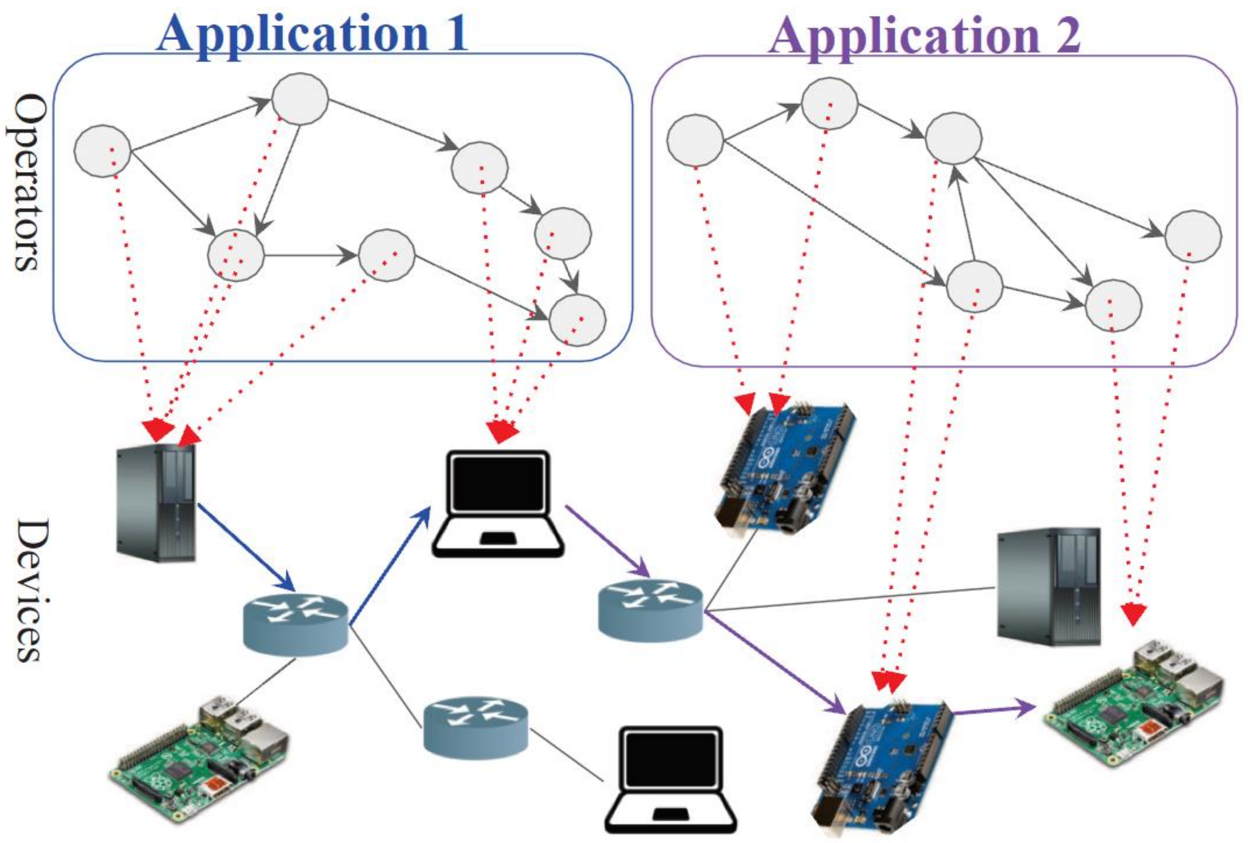
- Distributed Analytics is helpful for complex analytic applications
- Different cut points are suitable for different environments
  - i.e., while network resource is the bottleneck, we tend to put the cut point closer to device 2
- The overhead caused by Docker and TensorFlow are small
  - Docker virtualization overhead: < 5%
  - TensorFlow communication overhead: < 10%

Conclusion

- Implement a fog computing platform running distributed analytics applications
  - Dynamic Deployment -> Docker
  - Efficient Management -> Kubernetes
  - Distributed Analytics -> TensorFlow
- Implement real analytics applications
  - Air pollution monitor and object recognizer
- Experiments
  - Better performance of distributed analytics
  - Low overhead caused by Docker and TensorFlow
  - Tradeoff of different cut points

Future Work

- Future Work
  - Deployment problems
  - Available resource prediction
  - Seamless migration
  - …



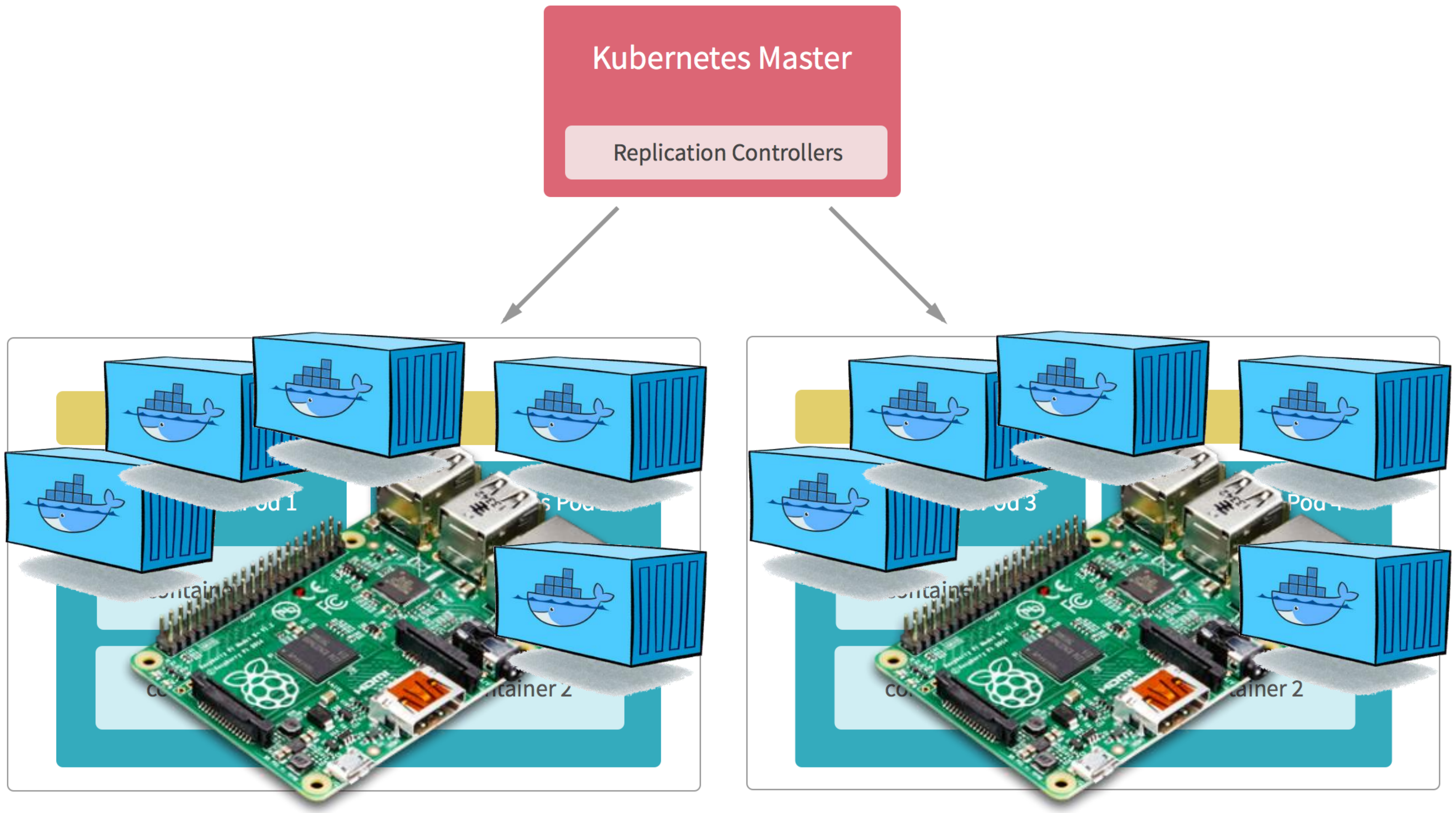
Q&A

Management

- OpenStack
  - Used to manage virtual machines in data centers
- SaltStack
  - Remote execution tool and configuration management system
- Kubernetes
  - Automating deployment, scaling, and management of containerized applications

Kubernetes Architecture

- Each hosts several containers, can be assembled into pod
- A service is a group of pods that are running on the cluster



Container

- Docker
  - Container virtualization
  - Need less storage space and less computing power

- Kubernetes
  - Automated container deployment, scaling, and management

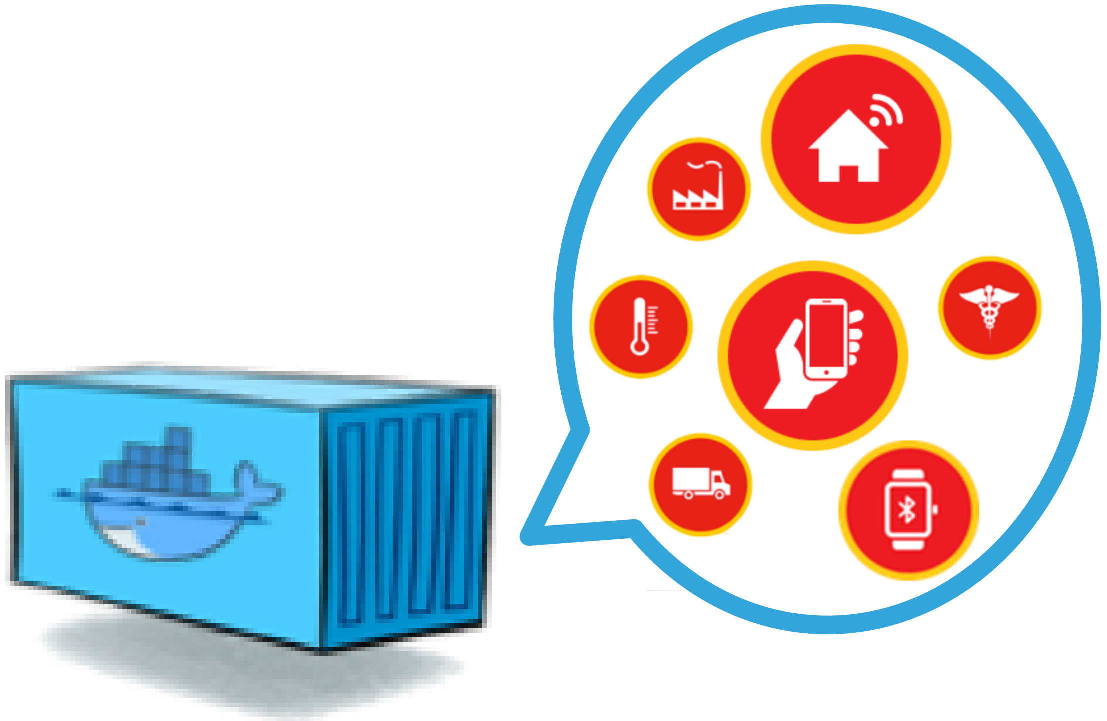
	Virtual Machine	Container
Size	GB	MB
Startup	Minute	Second

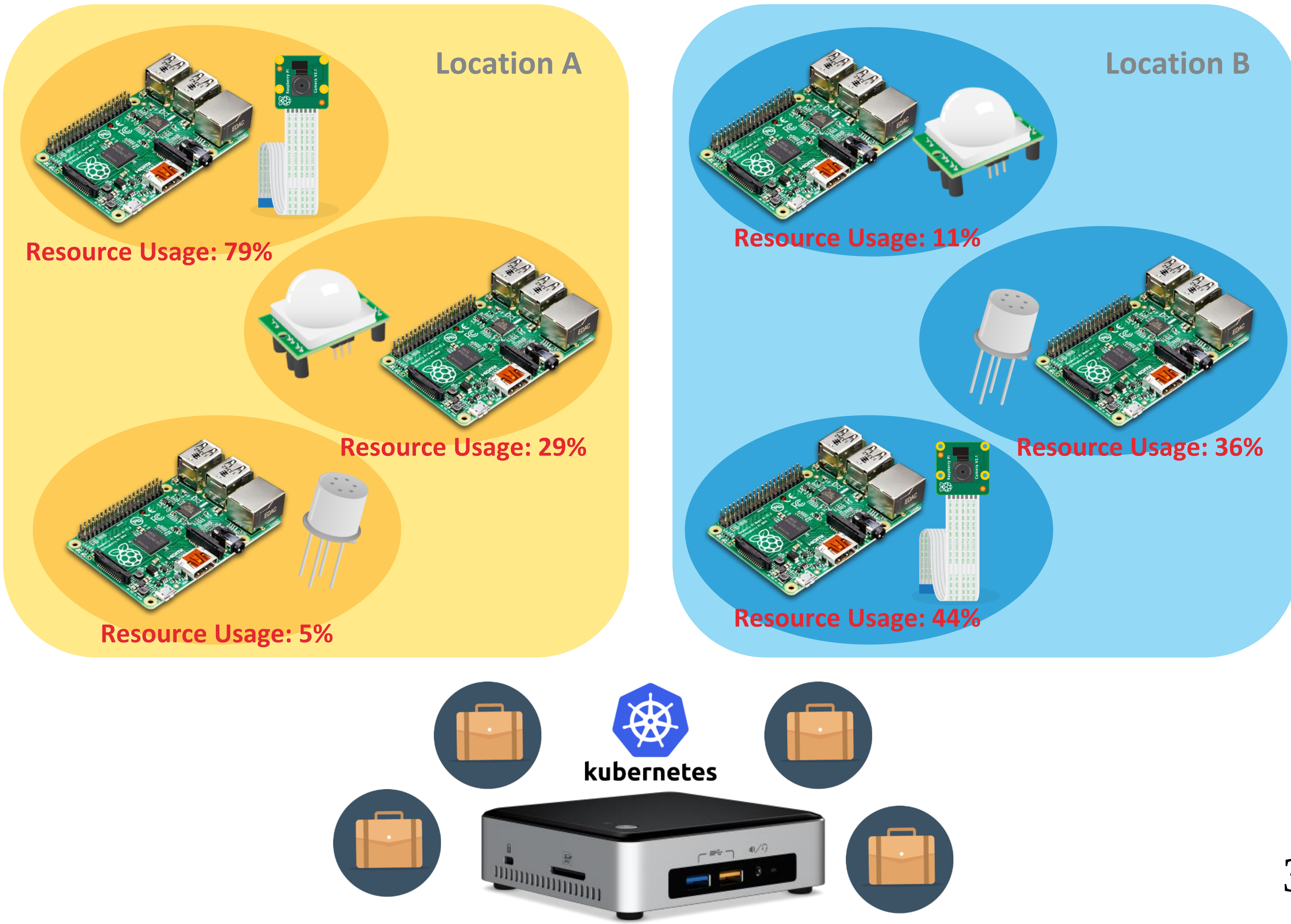
docker

kubernetes

Container-based Applications

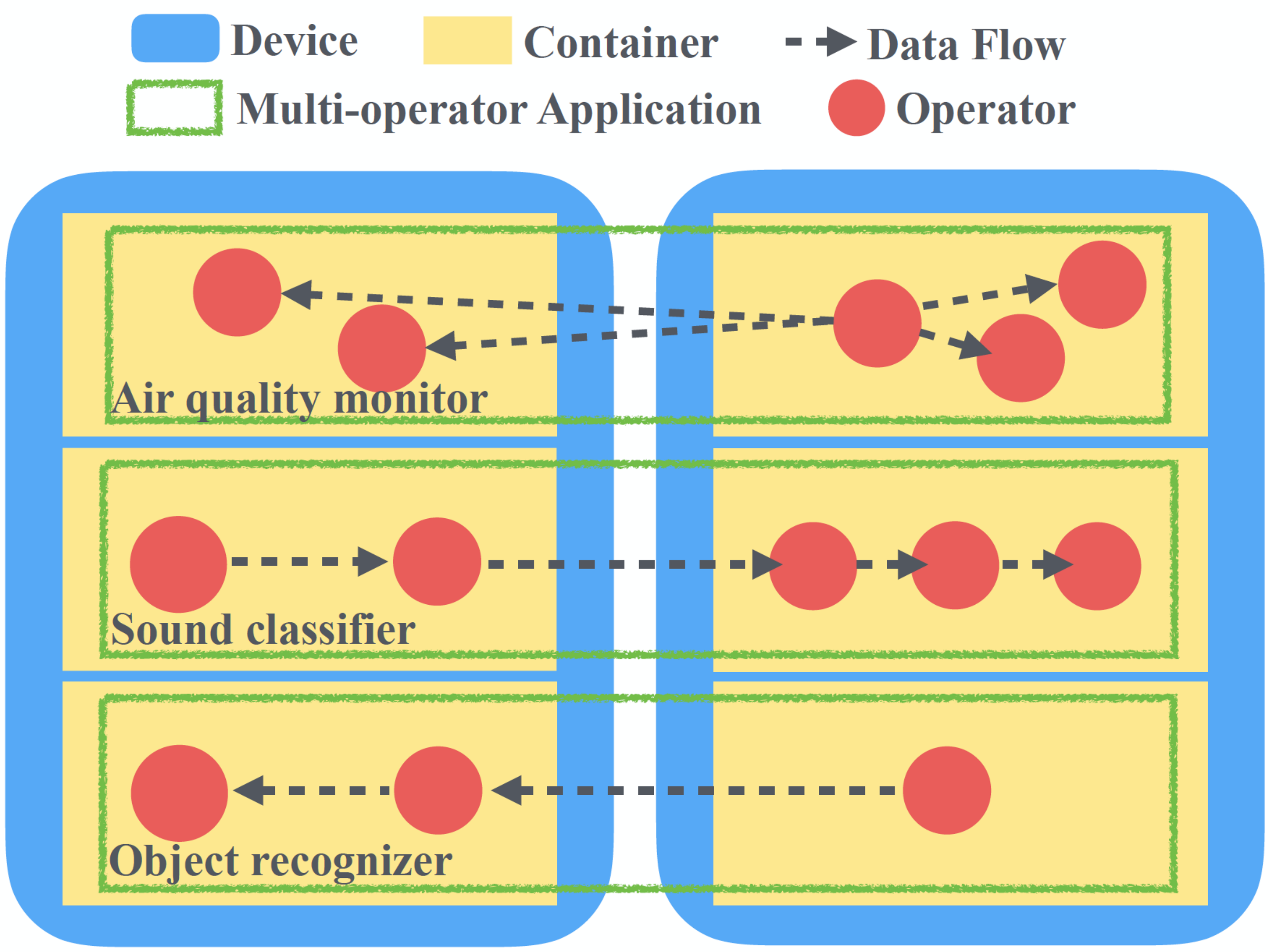
- Lightweight
  - Quick start
  - Easy to replace the configuration of the applications
- Each host can handle multiple containers
  - Share the same OS kernel, and use the namespaces to distinguish one from another

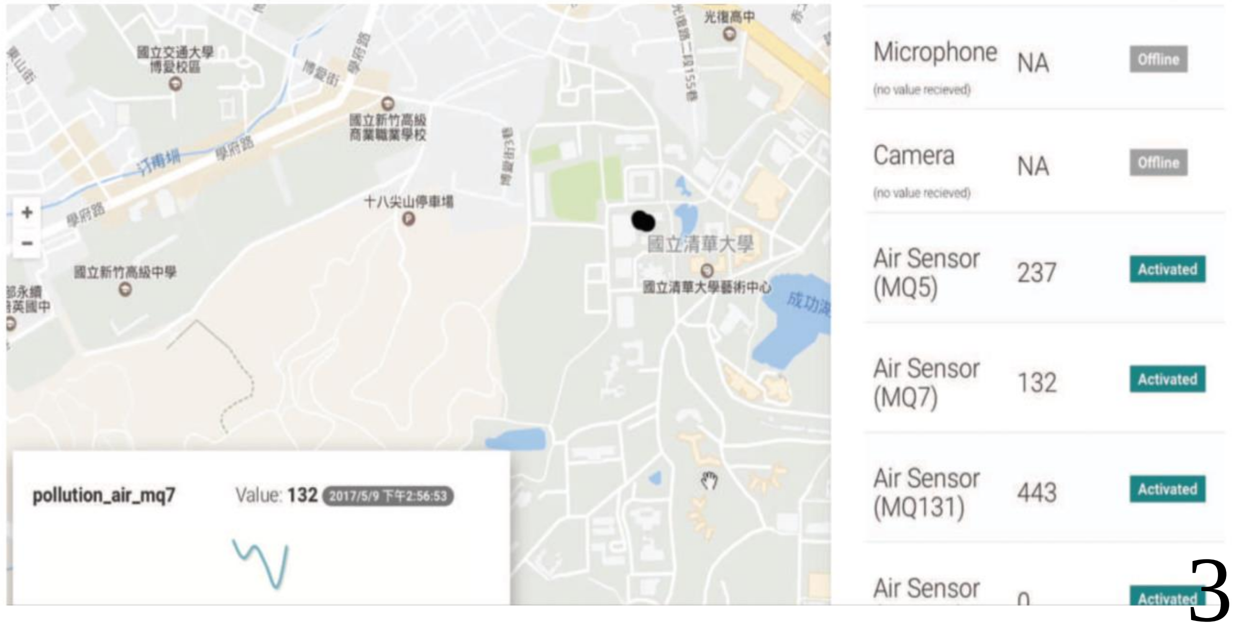



Location A
Resource Usage: 79%
Resource Usage: 29%
Resource Usage: 5%
Location B
Resource Usage: 11%
Resource Usage: 36%
Resource Usage: 44%
kubernetes

Applications

- IoT edge analytics
  - TensorFlow
  - 3 applications
    - Air pollution
    - Object detection
    - Sound classification





Sample Analytics Applications

- Air pollution monitor
- Object recognizer
- Sound classifier