Assessing the Suitability of MicroVM with AI application in Edge Computing

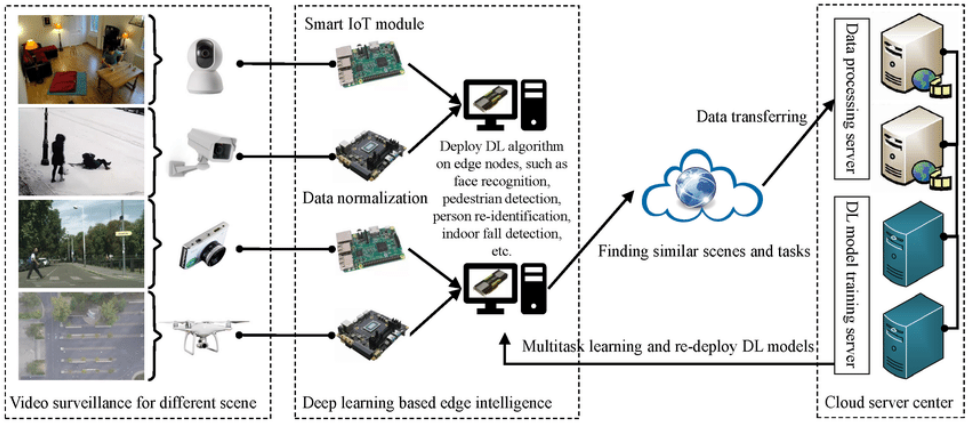
최윤하, 박준혁, 이경운, 탁병철

Kyungpook National University (KNU), Daegu, Republic of Korea

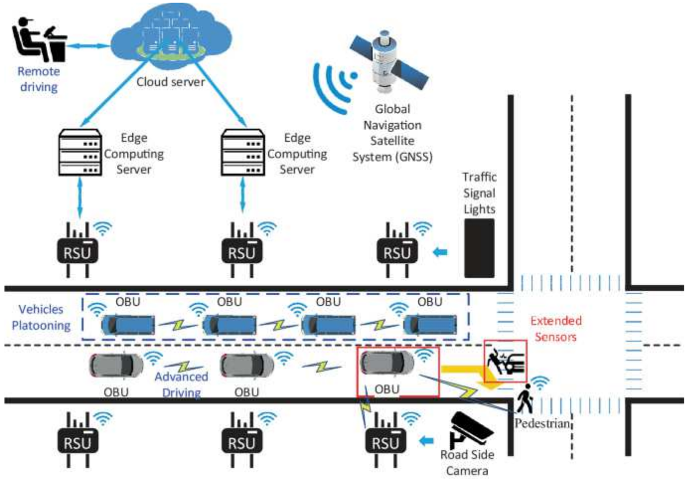
2024.July.09 Tuesday

KNU
KYUNGPOOK NATIONAL UNIVERSITY

Scenarios: Edge Computing for IoT


Smart IoT module
Data normalization
Deploy DL algorithm on edge nodes, such as face recognition, pedestrian detection, person re-identification, indoor fall detection, etc.
Data transferring
Finding similar scenes and tasks
Multitask learning and re-deploy DL models
Data processing server
DL model training server
Video surveillance for different scene
Deep learning based edge intelligence
Cloud server center

Surveillance Camera

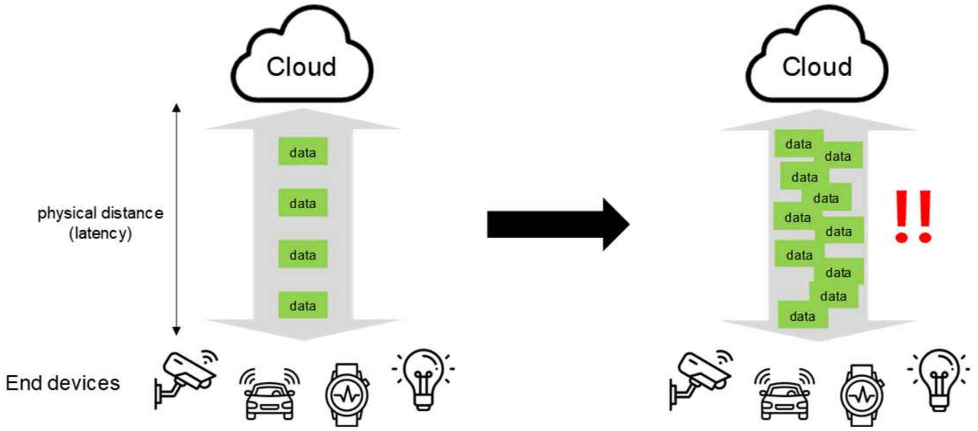


Autonomous vehicle

Challenges of Cloud Computing

- Cloud computing: Provide computing services such as servers, storage, networking, software, and analysis over the Internet

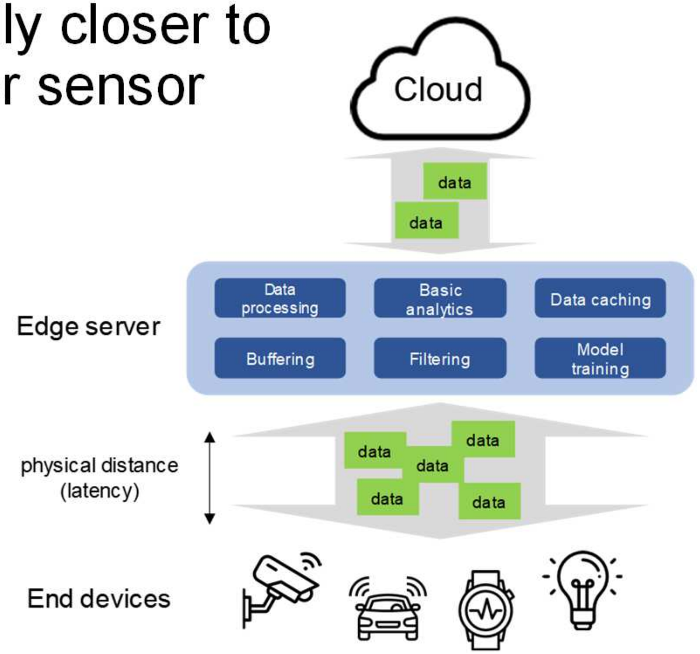
- Disadvantages
  - **High latency**
    - ► long physical distance to cloud data center causes delay
    - ► It is fatal for scenarios that require immediate response, such as surveillance camera, autonomous driving
  - **Increased bandwidth cost**
    - ► limited bandwidth to process big data
  - Data security and privacy
    - ► confidential and sensitive data should be given to cloud through network



Motivation 1: Edge Computing

- Edge computing: Move computing power physically closer to where data is generated, usually an IoT devices or sensor

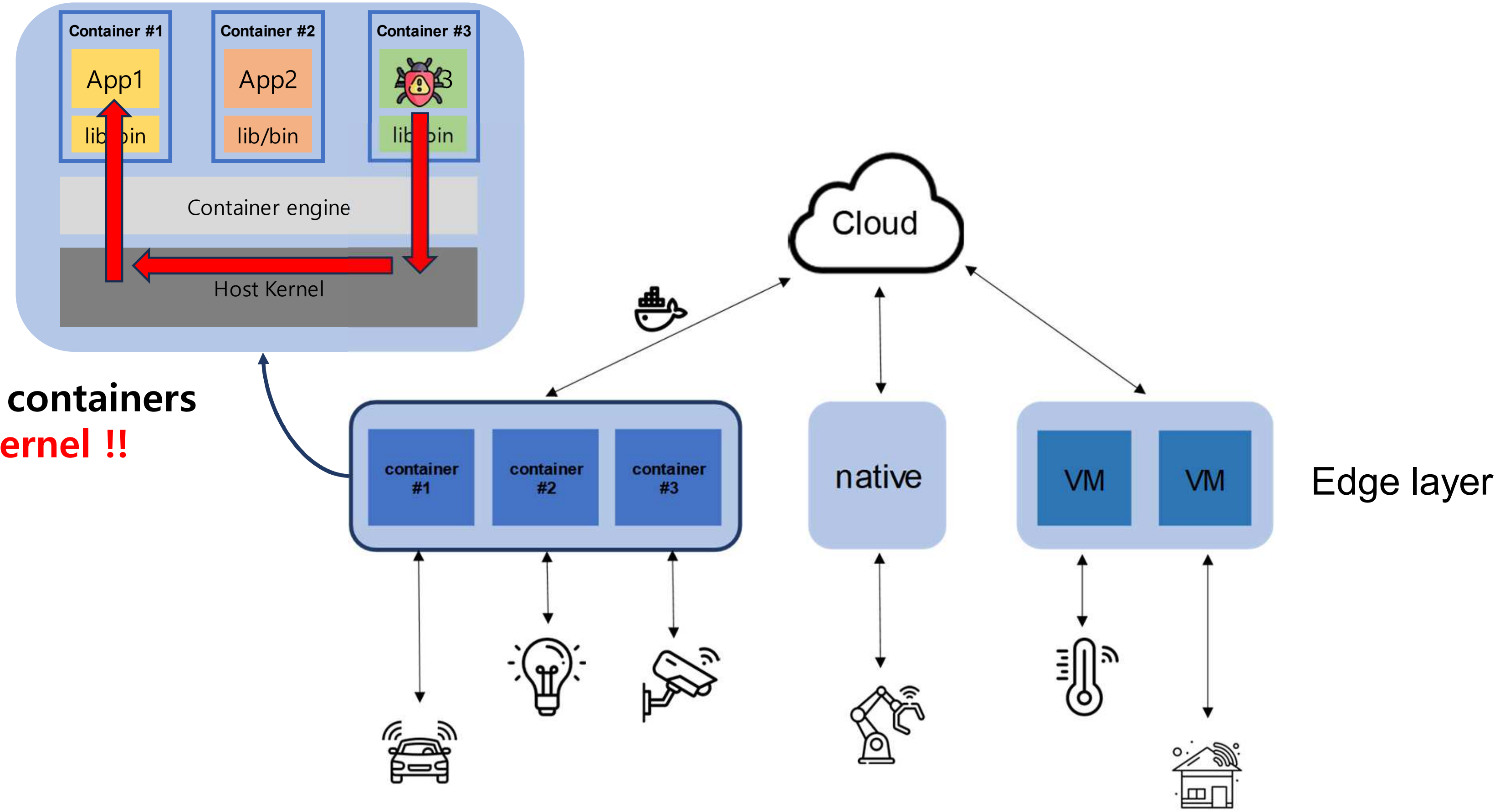
- What's better than cloud computing?
  - **Faster data processing**
  - **Low latency**
  - Reduced cost
  - Wider reach
  - Ensured data sovereignty



기존의 cloud computing은 대용량의 데이터를 처리하기에는 문제점이 존재한다.
따라서, edge computing을 활용하여 효율성을 높일 수 있다.

Benefits and Security Issue of Container

- Containerization is used when addresses many challenges of operating software systems at the edge

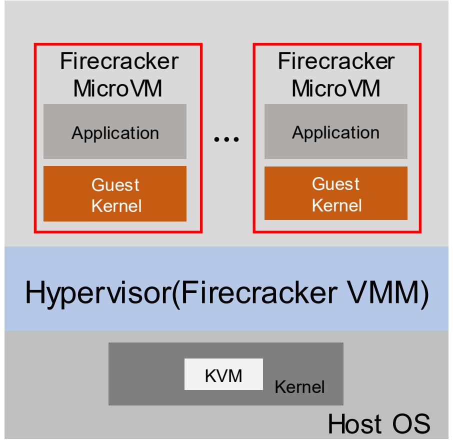


Fatal hazard of containers
→ Share host kernel !!

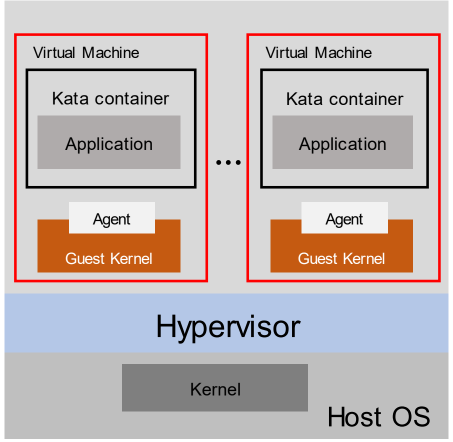
Motivation 2: The Rise of MicroVM

- MicroVM is **lightweight virtualization** technology which combines the advantages of container and VM
  e.g. AWS Firecracker, kata-container

- Big difference from container?
  - **Unnecessary guest kernel related functions are excluded**
  - It has its **own kernel**, so provide strong isolation
  - It needs hypervisor which manages VM



AWS Firecracker



Kata container

기존 컨테이너가 호스트 커널을 공유한다는 치명적인
보안 문제를 해결하기 위해 microVM이 등장하였다.

Summary of Motivations

- Edge devices have **limited computing resources**, and microVMs have an **optimized internal structure**, unlike containers and traditional virtual machines. Therefore, it is **difficult to predict the results** when performing AI applications.
- Edge device에서 가상화 환경을 구축한 후, AI application의 수행 성능 비교

→ **microVM이 기존의 가상화 환경들을 대체할 수 있는가?**

+

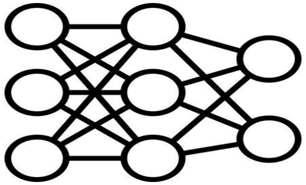
Docker container

AWS Firecracker

kata container

Linux KVM

+



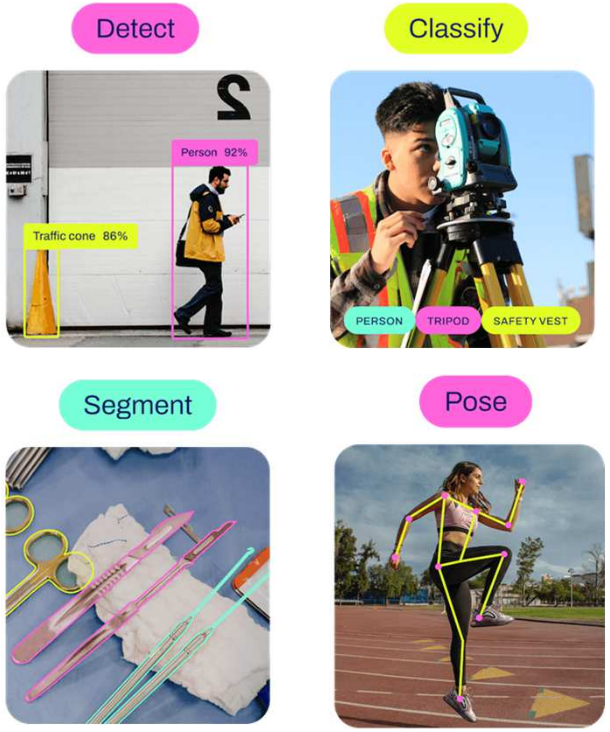
Experiment Design

- Edge device
  - Raspberry Pi 4(8GB RAM), NVIDIA Jetson Nano(4GB RAM)
- Virtualization environment
  - native, Docker container, AWS Firecracker, kata container, Linux KVM
- AI application: YOLOv8
- Test data: MS COCO

- **Check inference time & host CPU usage of each environment**

YOLOv8

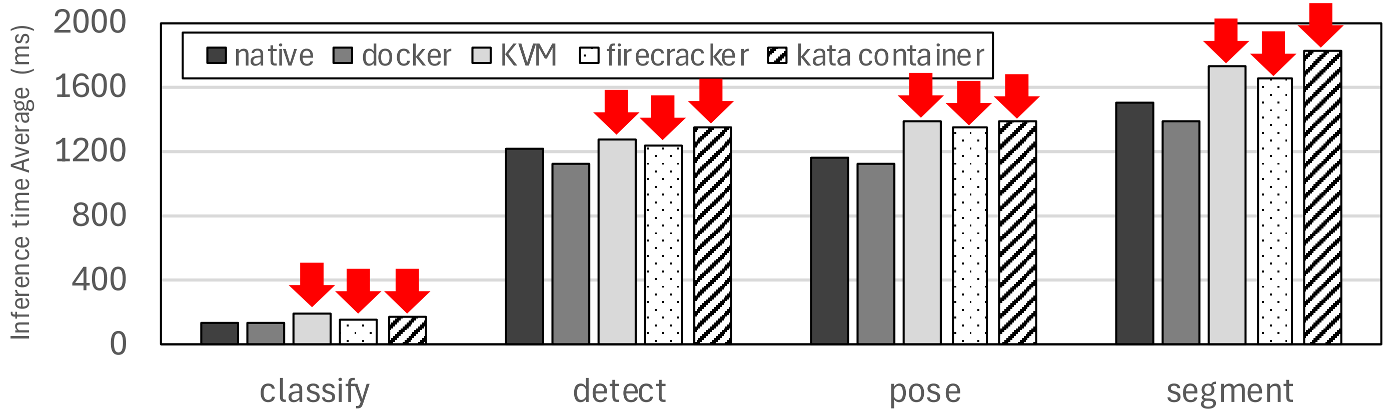
- New state-of-the-art computer vision model
- Provides 5 tasks and various modes (training, validation, prediction, etc.)
- Easy to use by installing Python package
- Many hyperparameters and configurations
  - batch size, learning rate, momentum, etc.
- Predict mode
  If we put test data to the model, we can get…
  - result image(e.g. with bounding box)
  - pre/post processing time
  - **inference time**



YOLOv8 tasks

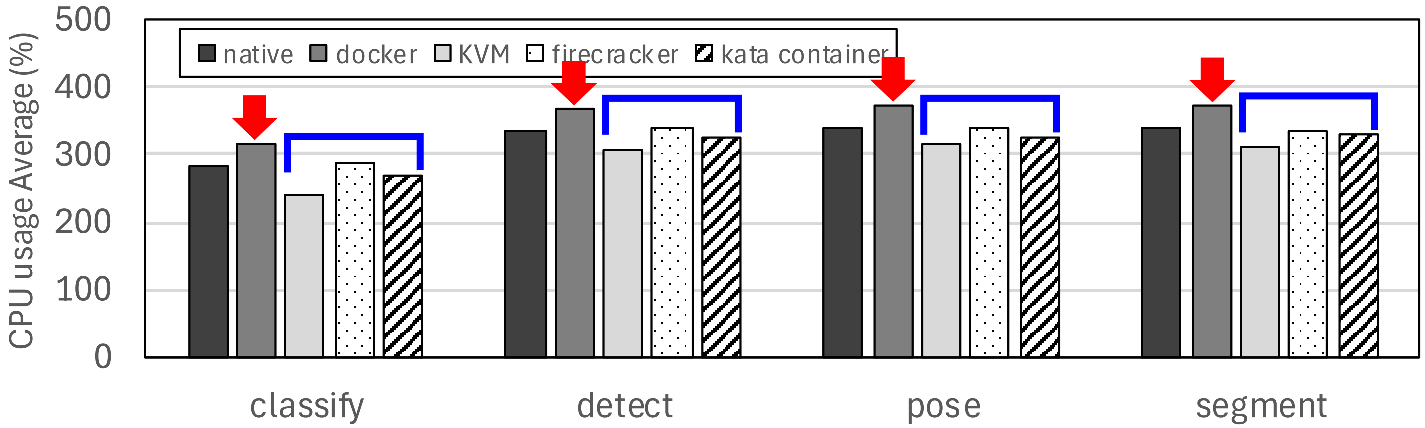
Experiment Result 1: Inference Time

Raspberry Pi 4



- Virtualization overhead such as hypervisor causes VM to show slower inference time
- Firecracker has improved performance by replacing existing hypervisor(i.e. QEMU) with optimized Firecracker VMM
- Kata container has more overhead because it uses double isolation structure for high-level security
- Containers that are not completely isolated from the host have high STD
  It may be difficult for developers to predict performance in real-world scenarios
- VM-based environments, which have their own isolated space, show relatively low STD

Experiment Result 2: CPU Usage



- Both Raspberry Pi 4 and Jetson Nano devices have 4 cores, the maximum is 400%
- Docker containers reduced inference time by increasing CPU usage to efficiently utilize host resources
- VM-based environments use less host CPU because guest VM abstract and manage hardware resources itself

Conclusion

- **Firecracker microVM** showed similar inference performance to the container
- **Kata container** is a little low in performance, but it provides better security with a double isolation layer
- There's no big difference between container and microVM in terms of AI application performance
- When using AI applications on the edge devices, it is expected that **microVMs can be used as an alternative to existing containers** depending on the user's purpose(security, performance, boot time, etc.)

- Limitations
  - MicroVM does not support GPU virtualization for now
  - Need more research on memory and network usage

QnA