A Smorgasbord of Embedded and Pervasive Computing Research

Kishore Ramachandran

(part of systems group which includes Ada Gavrilovska, Taesoo Kim, Ling Liu, Calton Pu, and Alexey Tumanov)

Georgia Tech College of Computing

- Current PhD inmates!

Enrique Saurez

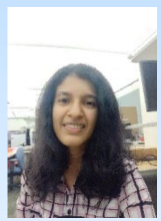
Harshit Gupta

Zhuangdi Xu

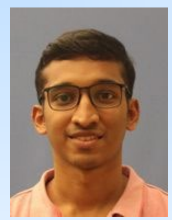
Adam Hall

Ashish Bijlani

Tyler Landle

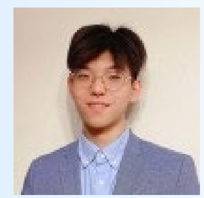


Manasvini Sethuraman



Anirudh Sarma

Alan Nussbaum



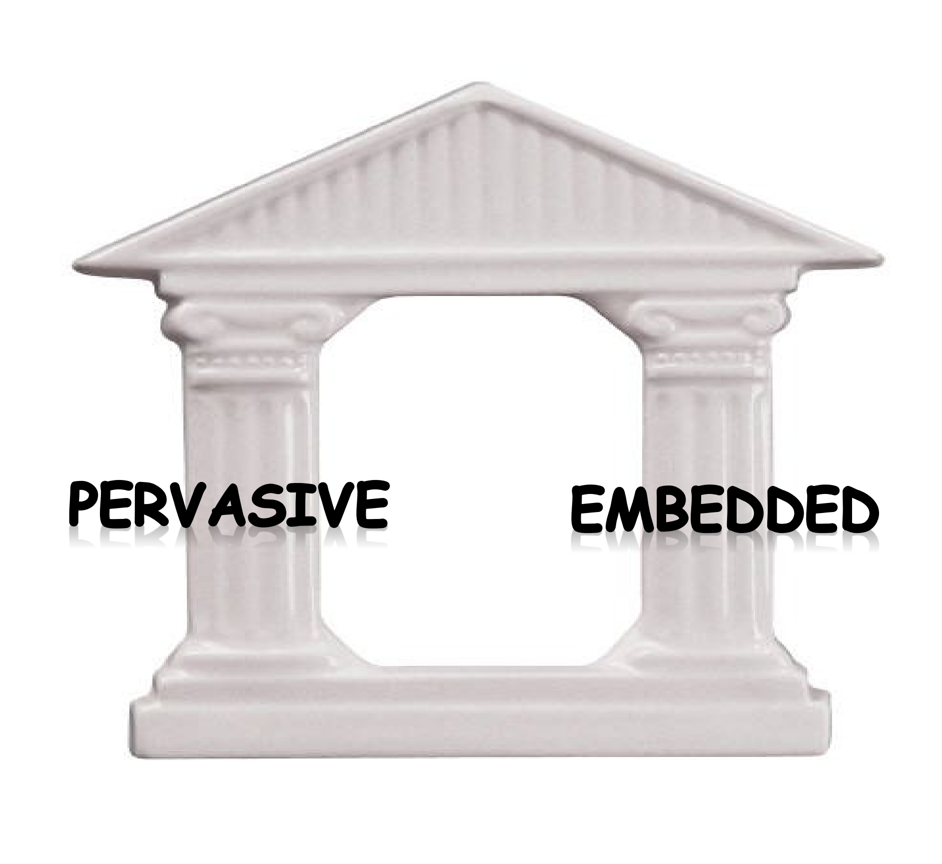
Difei Cao

Jinsun Yoo

- Recently escaped!
  - Hyojun Kim (IBM Almaden then Startup and now Google); Lateef Yusuf (Amazon then Google); Mungyung Ryu (Google); Kirak Hong (Google and now CTRL-labs); Dave Lillethun (Seattle U.); Dushmanta Mohapatra (Oracle); Wonhee Cho (Microsoft); Beate Ottenwalder (Bosch); Ruben Mayer (TU Munich), Ashish Bijlani (Startup)
- Plus a number of MS and UGs

Georgia Tech College of Computing

Embedded Pervasive Lab



Georgia Tech College of Computing

Pervasive side of the house

- Embedded devices treated as black boxes
- System Support for IoT
  - Fog/Edge computing

Current-Generation Applications

Cloud computing's utility model commoditized hardware…

Amazon
EC2

Microsoft
Azure

Google Cloud Platform

Enabling large-scale apps from centralized data centers…

facebook

NETFLIX

Next-Generation Applications

Future apps will be data-driven, model-driven, machine-in-the-loop, and far more demanding…

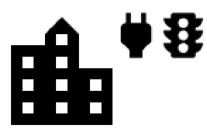
Autonomous Vehicles (AV)

Augmented / Virtual Reality (AR/VR)

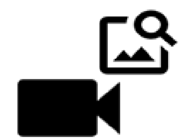
Latency <20 ms

Bandwidth >50 Mbps

Per device



Smart Cities



Smart Camera Networks

Next-Generation Applications

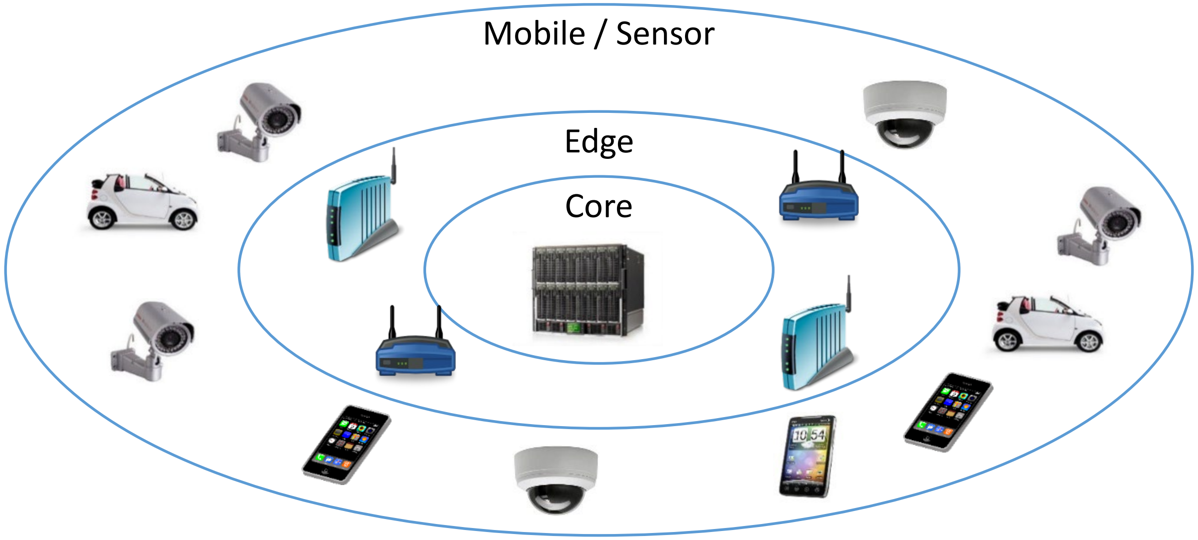
- Sense -> Process -> Actuate
- Common Characteristics
  - Dealing with real-world data streams
  - Real-time interaction among mobile devices
  - Wide-area analytics
- Requirements
  - Dynamic scalability
  - Low-latency communication
  - Efficient in-network processing

Cloud Computing

- Good for web apps at human perception speeds
  - Throughput oriented web apps with human in the loop
- Not good for many latency-sensitive IoT apps at computational perception speeds
  - sense -> process -> actuate
- Other considerations
  - Limited by backhaul bandwidth for transporting plethora of 24x7 sensor streams
  - Not all sensor streams meaningful
    => Quench the streams at the source
  - Privacy and regulatory requirements

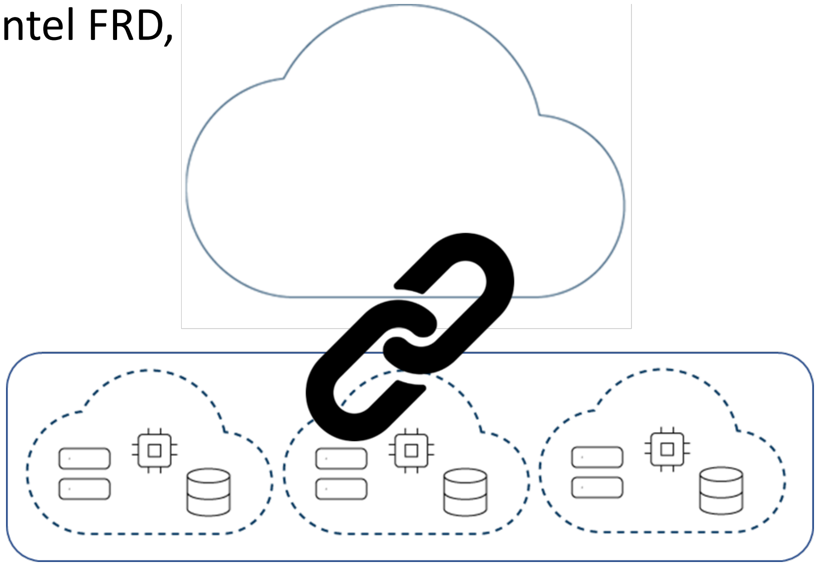
Fog/Edge Computing

- Extending the cloud utility computing to the edge
- Provide utility computing using resources that are
  - Hierarchical
  - Geo-distributed



Fog/edge computing today

- Edge is slave of the Cloud
  - Platforms: IoT Azure Edge, CISCO Iox, Intel FRD,
- Mobile apps beholden to the Cloud

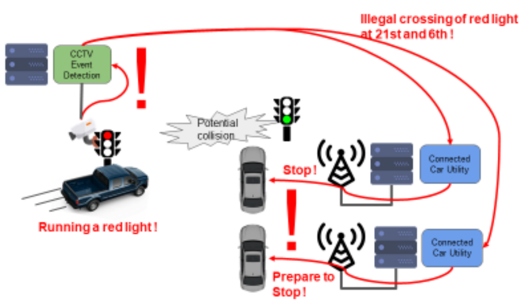


Vision for the future

- Elevate Edge to be a peer of the Cloud
  - Prior art: Cloudlets (CMU+Microsoft), MAUI (Microsoft)
- In the limit
  - Make the Edge autonomous even if disconnected from the Cloud

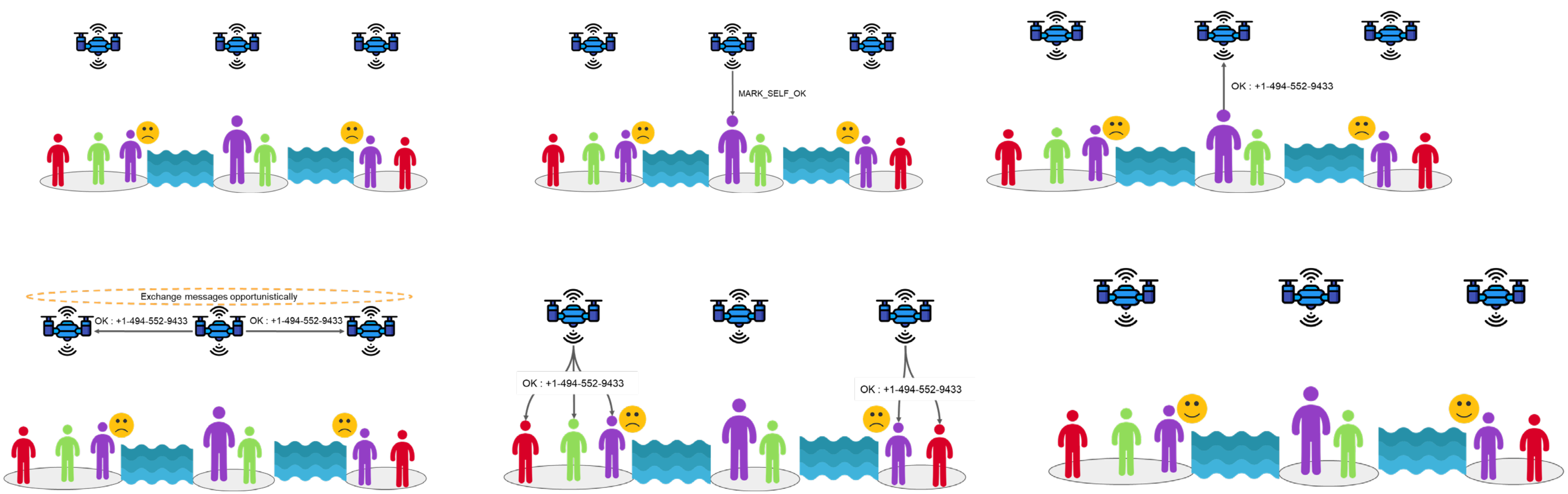
Why ?

- Interacting entities (e.g., connected vehicles) connected to different edge nodes
- Horizontal (p2p) interactions among edge nodes essential



Why ?

- Autonomy of edge (disaster recovery)



Challenges for making

- Need for powerful frameworks akin to the Cloud at the edge
  - Programming models, storage abstractions, pub/sub systems, …
- Geo-distributed data replication and consistency models
  - Heterogeneity of network resources
  - Resilience to coordinated power failures
- Rapid deployment of application components, multi-tenancy, and elasticity at the edge
  - Cognizant of limited computational, networking, and storage resources

Thoughts on Meeting the Challenges
(https://www.cc.gatech.edu/~rama/recent_pubs.html)

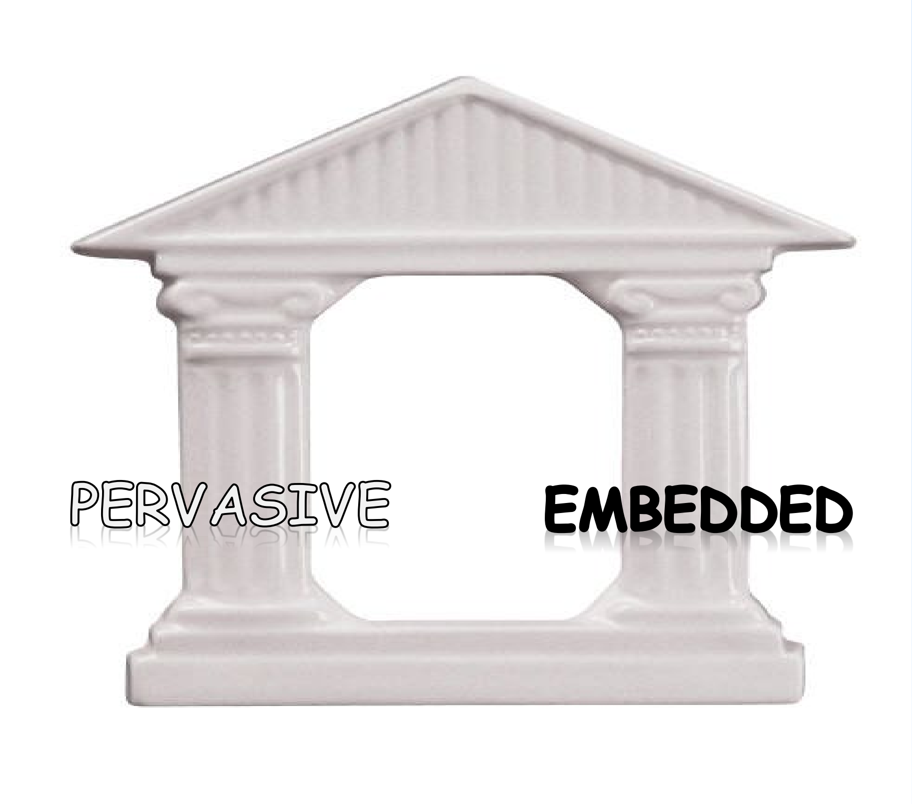
Theme: Elevating the Edge to be a peer of the Cloud

- Geo-distributed programming model for Edge/Cloud continuum
  - OneEdge (ACM SoCC 2021)
  - Foglets (ACM DEBS 2016)
- Geo-distributed data management – replica placement, migration and consistency
  - EPulsar (ACM SEC 2021)
  - FogStore (ACM DEBS 2018)
  - DataFog (HotEdge 2018)
- Efficient Edge runtimes
  - Serverless functions using WebAssembly (ACM IoTDI 2019)
- Applications using autonomous Edge
  - Social Sensing *sans* Cloud (SocialSens 2017)
  - STTR: Space Time Trajectory Registration (ACM DEBS 2018)
  - STVT: Space-Time Vehicle Tracking (HotVideoEdge 2019)
  - Coral-Pie: Space-Time Vehicle Tracking at the Edge (ACM Middleware 2020)
- Vision: “A case for elevating the edge to be a peer of the cloud”, GetMobile, 2020
- Vision: "eCloud: Vision for the Evolution of Edge-Cloud Continuum", IEEE Computer, 2021.

Ongoing Work

- eCloudSim (with **Daglis**, **Chatterjee**, **Dhekne**)
  - eCloud Simulation Infrastructure
- Prescient video prefetching at the edge for AV infotainment (With **Dhekne**)
  - Use route to JIT prefetching and caching for DASH player on vehicle
    - Foresight (ACM MMSys 2021)
  - Use mmWAVE (integrated with 5G LTE for edge node selection) to beam to passing vehicle
    - ClairvoyantEdge (in submission)
- Edge centric video data management systems for AV (With **Arulraj**)
  - Annotations with video for query processing, multi-tenancy, and sharing
    - EVA (To appear in ACM SIGMOD)
- Nimble execution environments for the Edge
  - Analyze cold start times in containers
  - Clean slate exec environment for FaaS
- NFSlicer: dataplane optimizations for processing network functions (With **Daglis**)
  - Selective data movement (e.g., header vs. payload) for NF chaining
    - NFSlicer (in submission)
- Performance isolation in the Edge (with **Sameh Elnikety** – Microsoft)
  - Harvest Container: Mixing latency-sensitive and throughput-oriented apps at the edge
- Hardware accelerators in micro-datacenters (with **Tushar Krishna**)
  - Efficiently using accelerators to improve efficiency in edge datacenters.

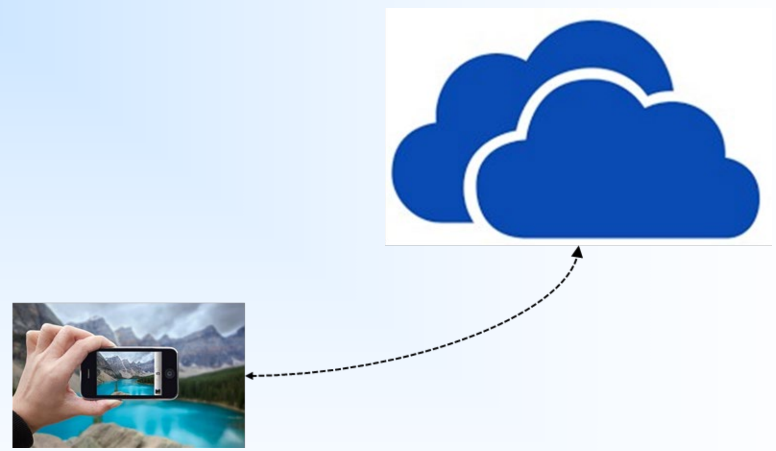
Embedded side of the house



- Infinite storage for mobile devices
- Optimizing Mobile Video Downloads

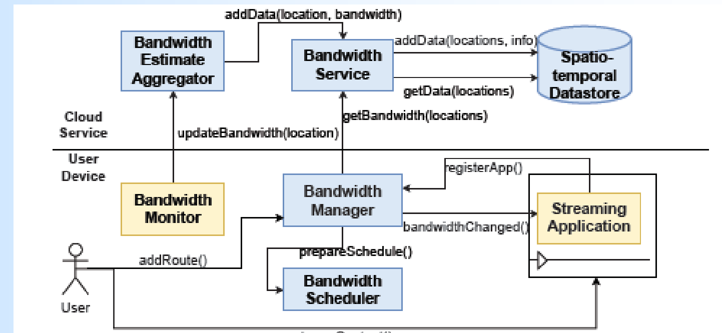
Infinite Storage for Mobile Devices

- Seamlessly extend the storage on mobile to the Cloud for any app
  - User space file system
    - APSys 2018, USENIX ATC 2019, Sigmetrics 2021
- Use machine learning to build user's everyday working set and (off)load (un)wanted data
- Issues
  - Latency
  - Energy consumption
  - Security and privacy

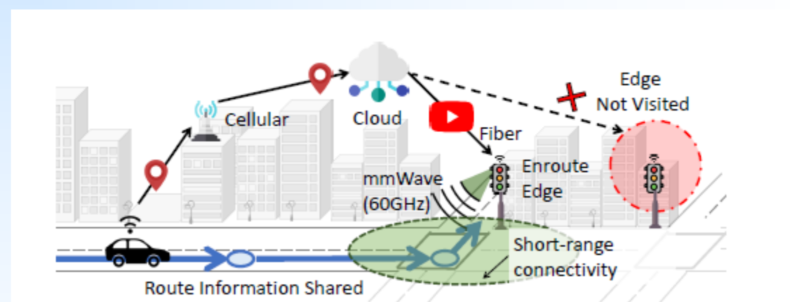


Optimizing Mobile Video Downloads

- Foresight (ACM MMSys 2021)
  - Bandwidth prediction across space and time for mobile users



- ClairvoyantEdge
  - Short range mmWave augmentation at Edge for high bandwidth video delivery



Georgia Tech College of Computing

Recap

- Infinite storage for mobile devices
- Optimizing Mobile Video Downloads
  - Foresight, ClairvoyantEdge

- Fog/Edge computing
  - eCloud
  - Foglets, OneEdge, thin virtualization for FaaS
  - Fogstore, DataFog, ePulsar, NFSlicer
  - STTR, Socialsens

Ongoing Projects

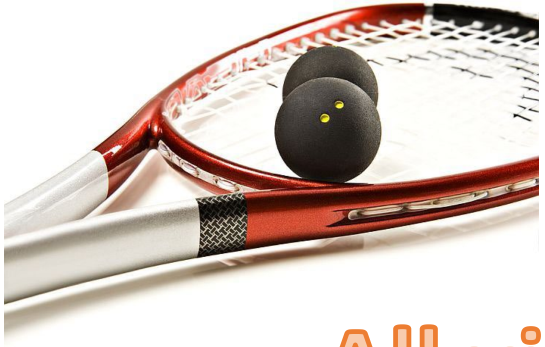
- **eCloud**: Device-Edge-Cloud continuum
- **OneEdge**: Device/Edge/Cloud control plane using AV as exemplar
  - Scheduling edge resources, monitoring, migration
- **Foresight** and **ClairvoyantEdge**: Prescient video prefetching at the edge for AV infotainment (With Prof. **Dhekne**)
  - Use route to JIT prefetching and caching for DASH player on vehicle
  - Use mmWAVE (integrated with 5G LTE for edge node selection) to beam to passing vehicle
- Edge centric video data management systems for AV (With Prof. **Arulraj**)
  - Annotations with video for query processing, multi-tenancy, and sharing
- Nimble execution environments for the Edge
  - Analyze cold start times in containers
  - Clean slate exec environment for FaaS
- **NFSlicer**: dataplane for processing network functions (With Prof. **Daglis**)
  - Selective data movement (e.g., header vs. payload) for NF chaining
- **MicroEdge**: Low-cost edge architecture for camera processing (With Prof. **Krishna**)
- Edge computing solution for underserved communities
  - Smart information services without WAN connectivity

Pubs:
http://www.cc.gatech.edu/~rama/recent_pubs.html

E-mail: rama@cc.gatech.edu

Lab:
http://wiki.cc.gatech.edu/epl

What does Kishore "really" do in his copious spare time when he is not teaching?



Squash anyone?

All virtual!

Soon real?

Table tennis anyone?

What you should take away?

- "Kishore" rhymes with "sea shore"
- Squash/Table-tennis
- EPL
- Fog/Edge computing
- Infinite storage on mobile/EdgeCaching

https://www.dreamstime.com/namaste-vector-hand-drawn-symbol-yoga-design-image112101574

Georgia Tech College of Computing

Georgia Tech College of Computing

Extra slides on EPL projects underway

Kishore Ramachandran

Georgia Tech College of Computing

Two topics

Camera Processing @ Edge

Serverless @ Edge

Camera Network

Context

- Streaming 24 x 7
- Used by multiple applications
- Processing at ingestion time
- Processing at Edge

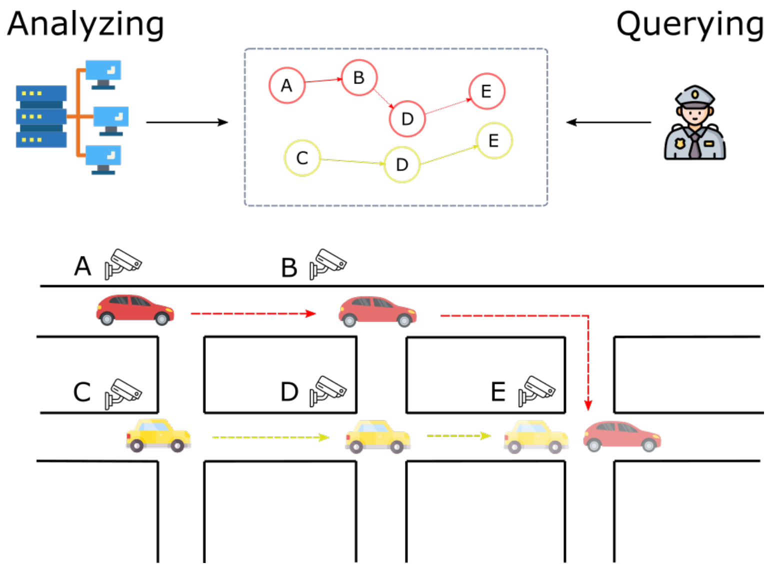
Applications

- Suspicious vehicle tracking
- Smart traffic light
- Camera-assisted navigation for the disability

Motivating Application

Space-Time Vehicle Tracking at video ingestion time

- Manual Checking
  - Labor-intensive
  - Error-prone due to lapses in attention
- Intelligent systems
  - Replace manual labor
  - Aid human decision-making



Outline

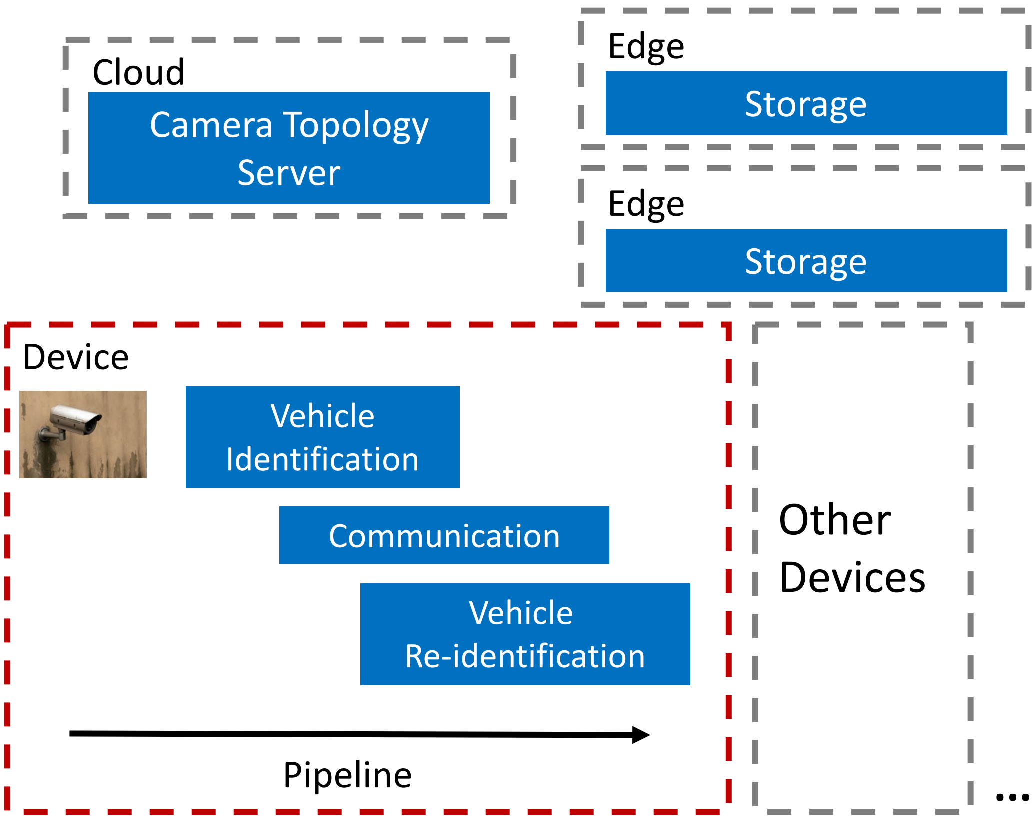
- Recent work on camera applications at Edge
  - Coral-Pie: Space-Time Vehicle Tracking at the Edge (**ACM Middleware 2020**)
- Proposed Work in a nutshell
  - Camera Networks at the Edge
    - MicroEdge: multi-tenancy edge system architecture
    - Accelerator orchestration in MicroEdge
  - Serverless at the Edge
    - Nimble orchestration for FaaS
    - Cross-site load balancing
    - Location-aware auto-scaling

Recent Work

Novel Principles

- "Scalable-by-design" camera processing at ingestion time
  - Per-camera latency-based subtask placement
  - Bounded network communication per camera
- Fault tolerance for geo-distributed camera network
  - Automatic camera topology management

Coral-Pie: Device-Edge-Cloud Continuum



Device:

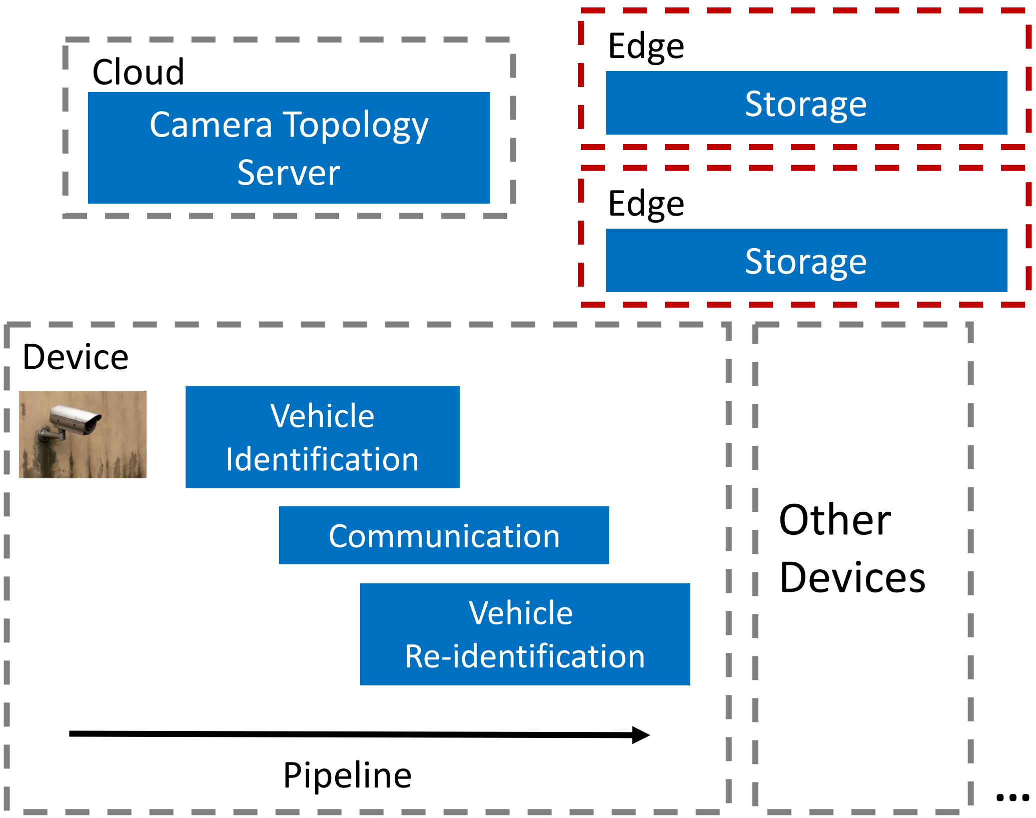
- Camera + 2 RPis + TPU

Heavy-lifting real-time camera processing

- **Deterministic latency bounds**
- **No pressure on the backhaul network bandwidth**

Pipelined processing to exploit parallelism

Coral-Pie: Device-Edge-Cloud Continuum



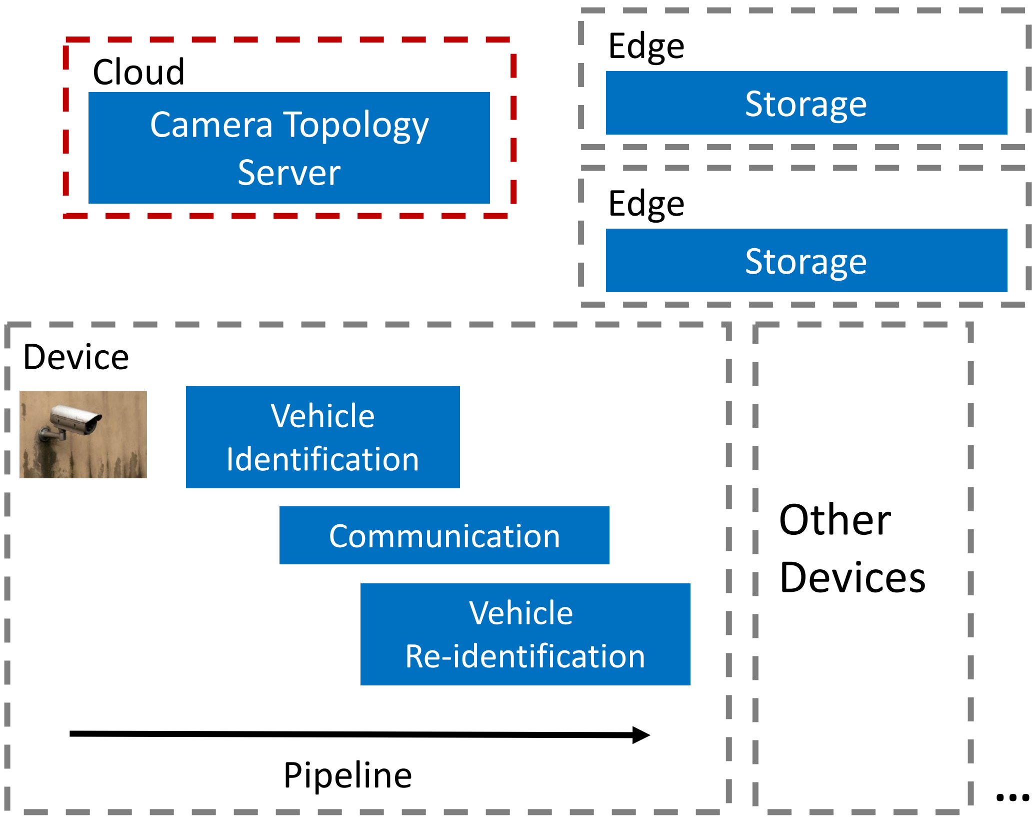
Edge:

- A multi-tenant micro-datacenter housed in a small footprint location
- **Few network hop(s)** away from Devices

Offloading storage

- ***Asynchronously***
- Helps balance the pipeline of tasks on the Devices
- Backhaul wide-area network (WAN) is not pressured

Coral-Pie: Device-Edge-Cloud Continuum



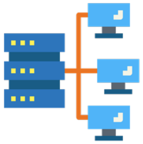
Cloud

- A centralized service provider that all devices can connect to
- **Global knowledge**

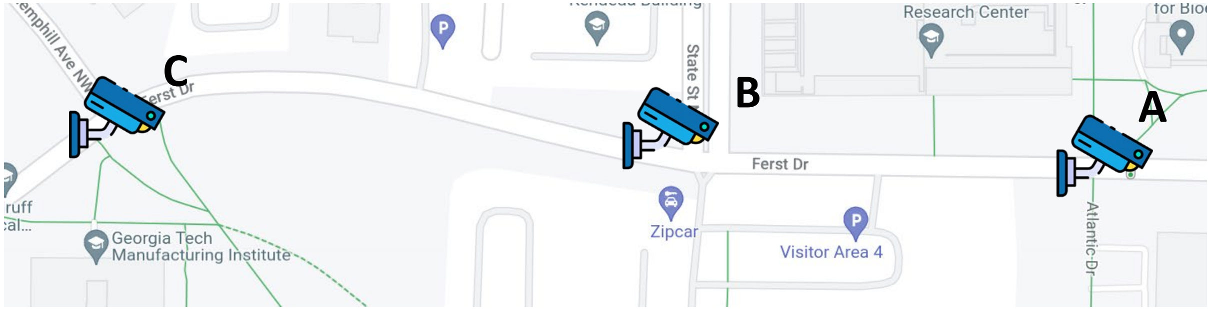
Maintains geographical relationship between the cameras

- Infrequently updated (i.e., a new camera is deployed or an old camera is removed)

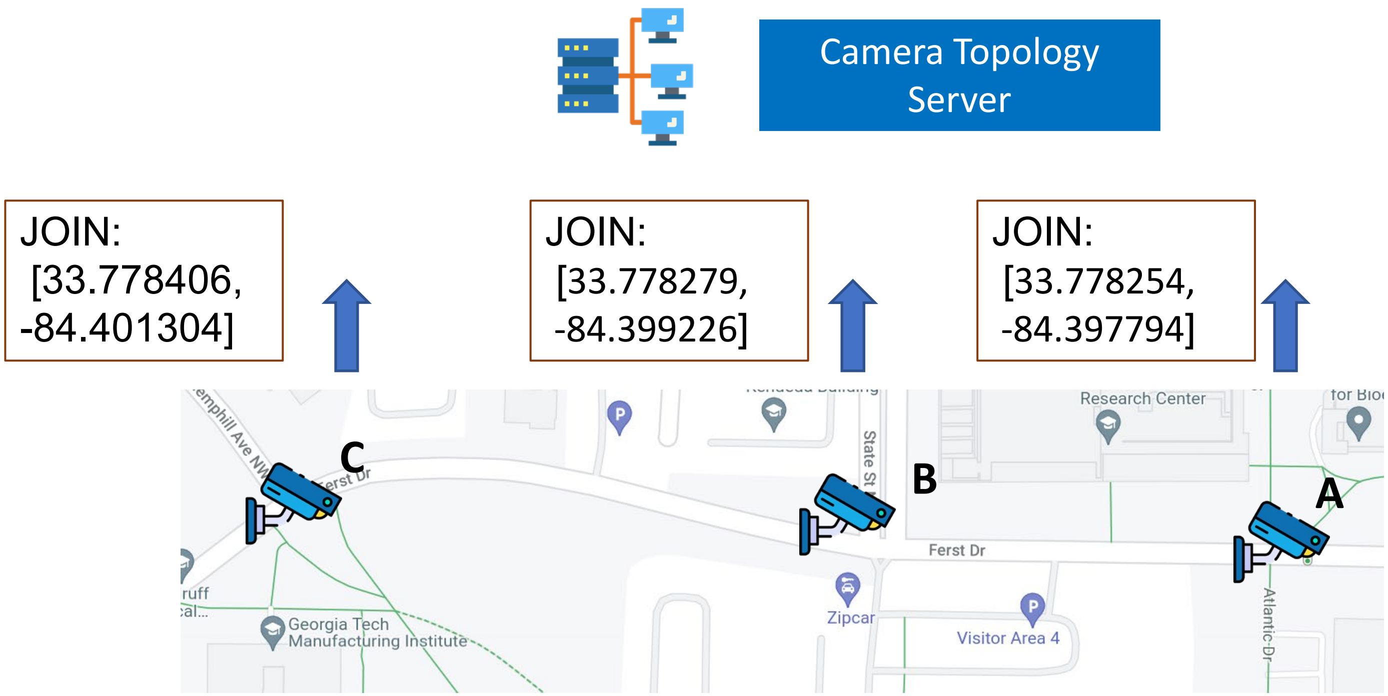
Coral-Pie in Action



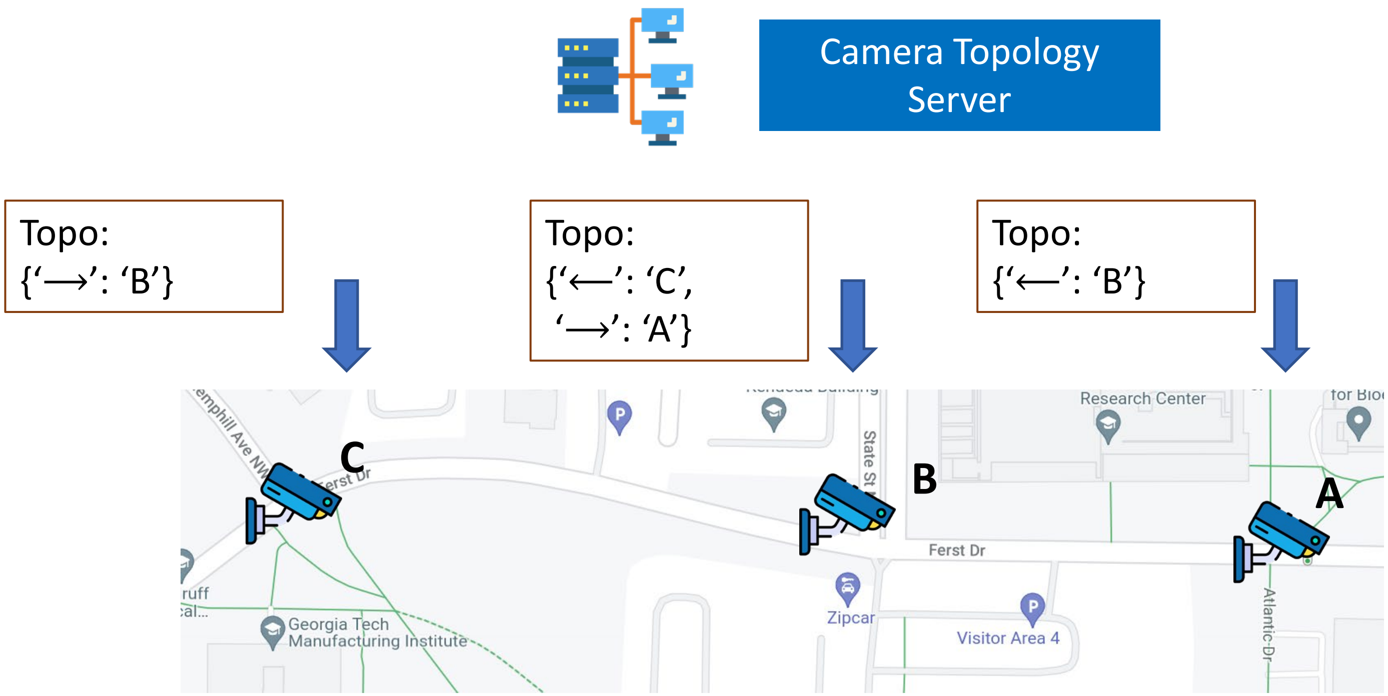
Camera Topology Server



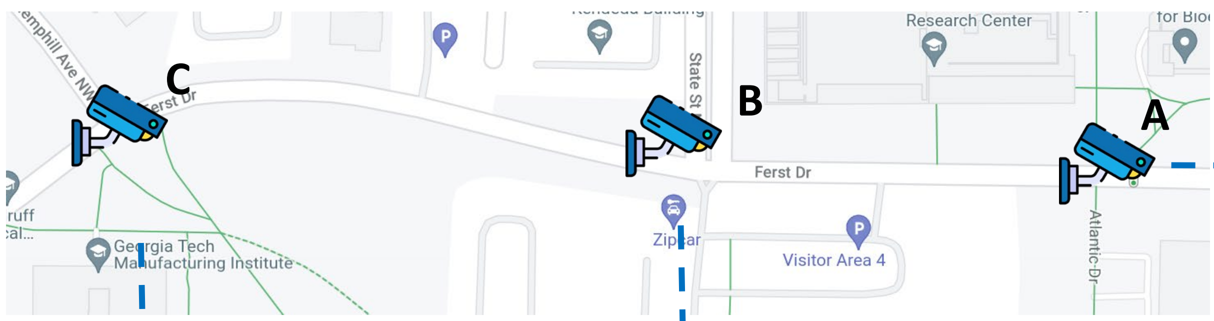
Cloud: Management of Camera Topology



Cloud: Management of Camera Topology

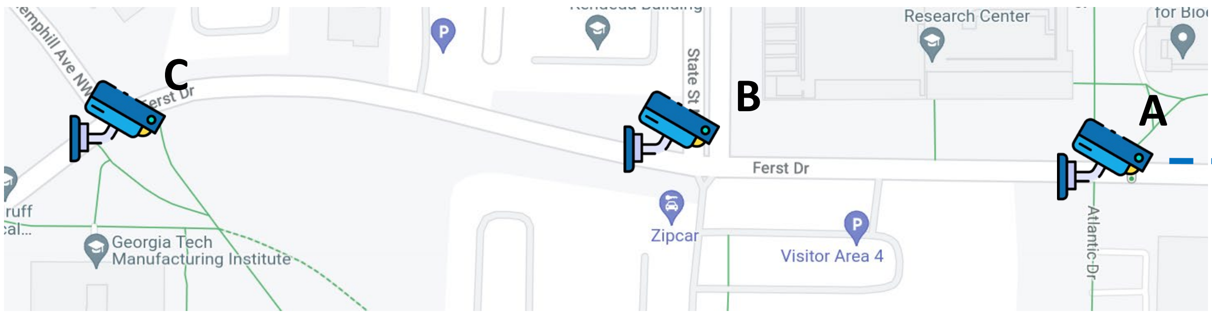


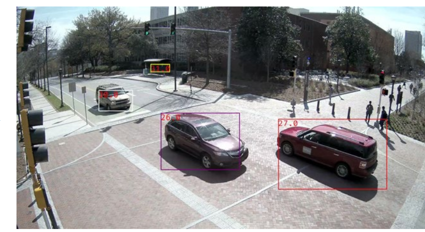
Device: Vehicle Identification



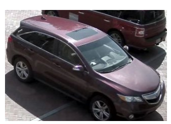
```
DetectionEvent_A_0:
{
    camera: A,
    timestamp: 18:19-07/10/2019,
    tracklet: [(bbox, frameId),...],
    features: {
        moving_direction: 270 (⟵),
        histogram: np.array([...])
    }
}
```

Device: Communication



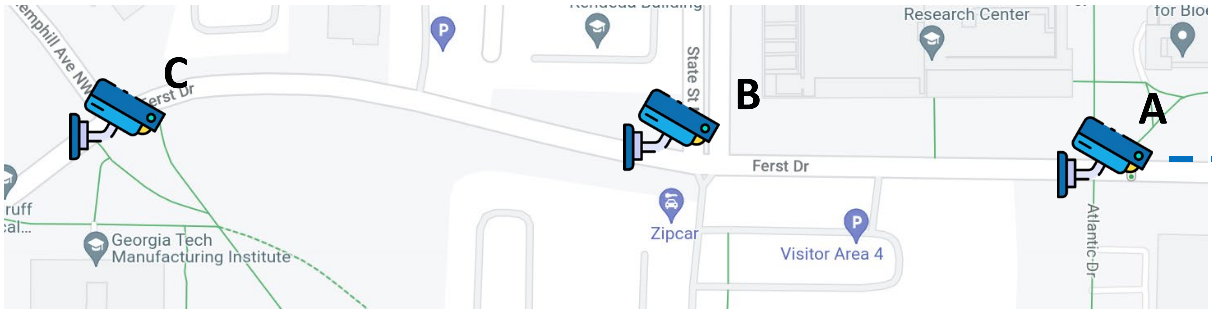


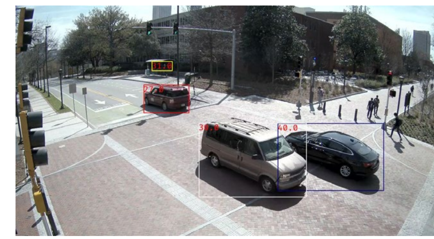
Camera B's Candidate Pool:

DetectionEvent_A_0: 

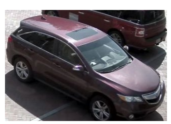
DetectionEvent_A_0:
{
    camera: A,
    timestamp: 18:19-07/10/2019,
    tracklet: [(bbox, frameId),...],
    features: {
        moving_direction: 270 (⟵),
        histogram: np.array([...])
    }
}

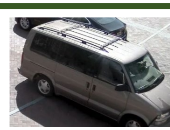
Device: Communication





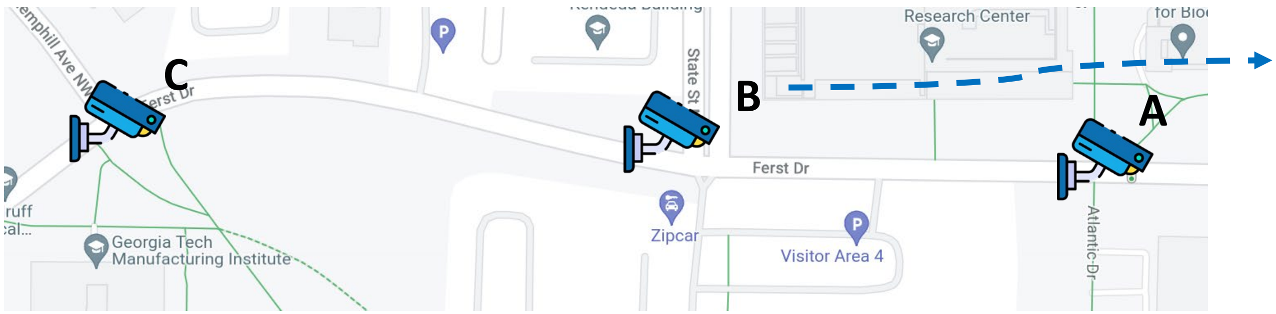
Camera B's Candidate Pool:

DetectionEvent_A_0: 

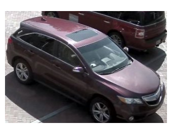
DetectionEvent_A_2: 

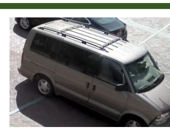
DetectionEvent_A_2:
{
camera: A,
timestamp: 18:19:09-07/10/2019,
tracklet: [(bbox, frameId),...],
features: {
moving_direction: 270 (⟵),
histogram: np.array([...])
}
}

Device: Vehicle Re-Identification at Camera B



Camera B’s Candidate Pool:

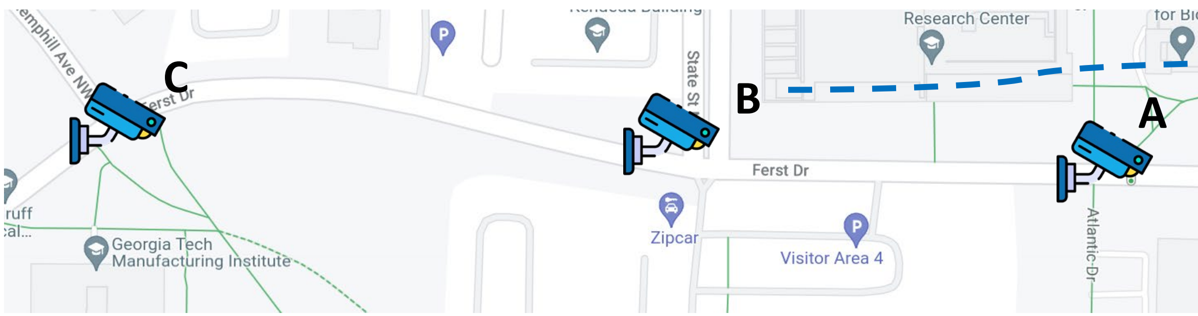
DetectionEvent_A_0: 

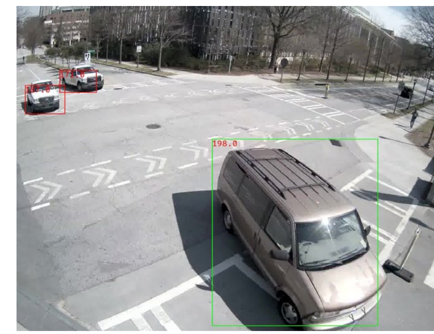
DetectionEvent_A_2: 

Matching!

```
DetectionEvent_B_7:
{
   camera: B,
   timestamp: 18:19:26-07/10/2019,
   tracklet: [(bbox, frameId),...],
   features: {
      moving_direction: 270 (⟵),
      histogram: np.array([...])
   }
}
```

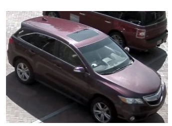
Device: Vehicle Re-Identification at Camera B



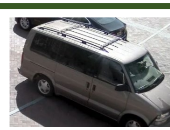


Camera B's Candidate Pool:

DetectionEvent_A_0:



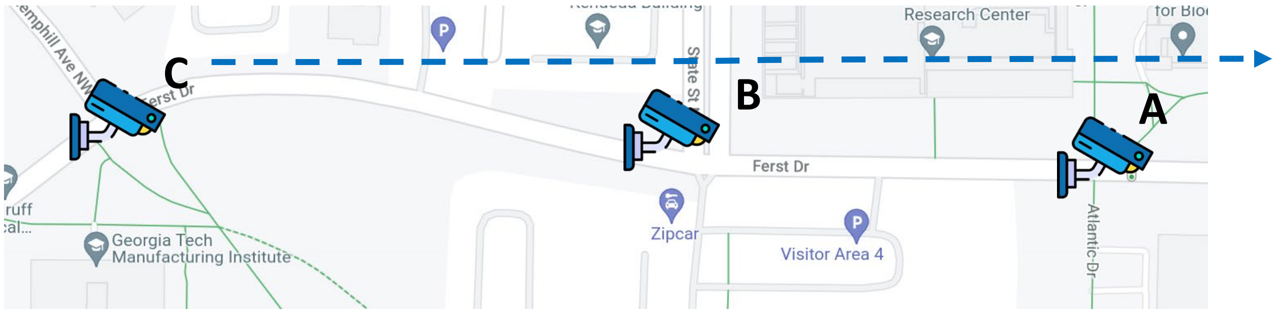
DetectionEvent_A_2:



Matching!

```
DetectionEvent_B_9:
{
   camera: B,
   timestamp: 18:19:29-07/10/2019,
   tracklet: [(bbox, frameId),...],
   features: {
      moving_direction: 180(↓),
      histogram: np.array([...])
   }
}
```

Device: Vehicle Re-Identification at Camera C



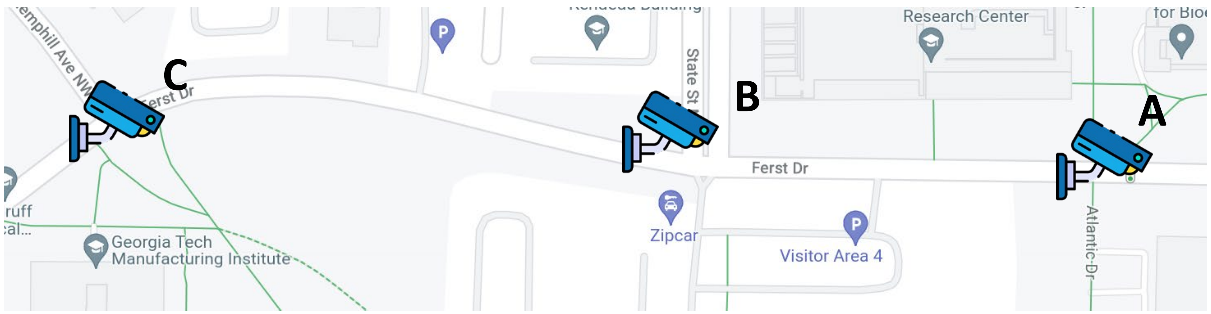
Camera B's Candidate Pool:

DetectionEvent_B_7: 

Matching!

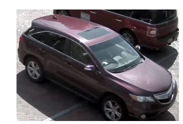
```
DetectionEvent_C_12:
{
   camera: C,
   timestamp: 18:19:46-07/10/2019,
   tracklet: [(bbox, frameId),...],
   features: {
      moving_direction: 315 (↑),
      histogram: np.array([...])
   }
}
```

Edge: Storage



- Trajectory Storage as Graph
- Frame Storage - traceability

DetectionEvent_C_12 ← DetectionEvent_B_7 ← DetectionEvent_A_0



DetectionEvent_B_9 ← DetectionEvent_A_2

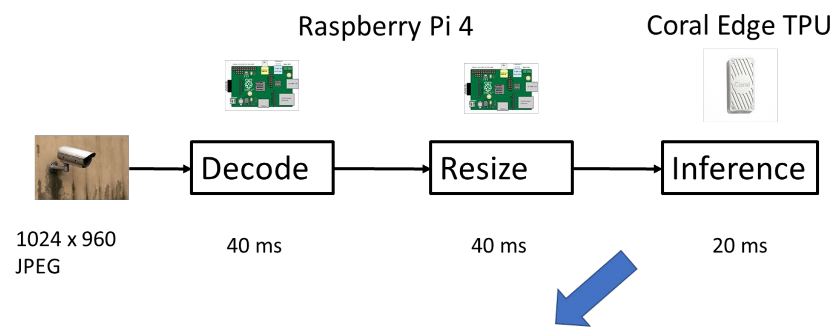
From Coral-Pie to MicroEdge

Limitations of Coral-Pie

- Dedicated hardware resources for each camera processing pipeline
  - ➢ Each pipeline may not use the resources all the time
- Camera network intended for multiple applications

Opportunity

- Share TPU, CPU, and memory resources within and across applications



- 50% TPU utilization with 25 FPS stream
- 20% TPU utilization with 10 FPS stream

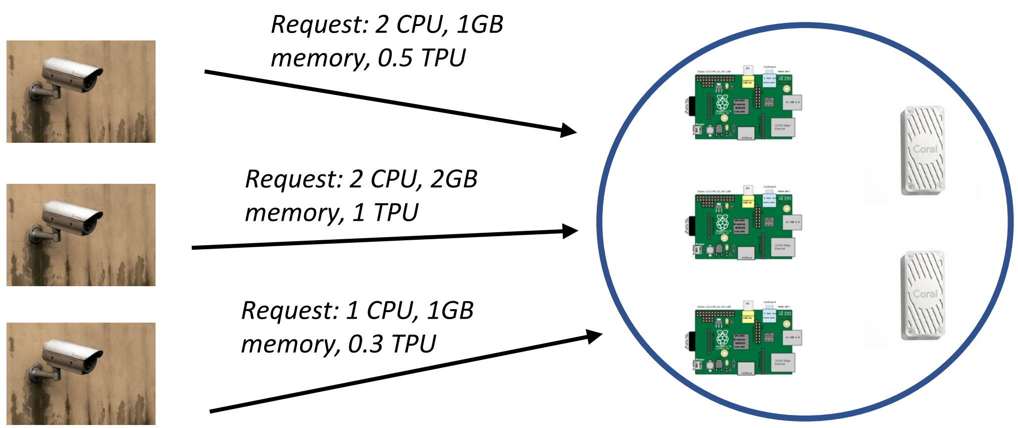
Limitations of the State-of-the-art

- Prior art focus on GPU management in resource-rich clusters
  - NVIDIA Docker and NVIDIA MPS
  - Clockwork[1], Heimdall[2], Infaas[3], and Clipper[4]
- Lack of support for multiplexing TPU resources across containers
- Missed opportunity in container-orchestration systems (e.g., Kubernetes, K3s)
  - Under-utilization of TPU resources

[1] Gujarati, Arpan et al. "Serving DNNs like Clockwork: Performance Predictability from the Bottom Up." OSDI (2020).
[2] Yi, Juheon and Youngki Lee. "Heimdall: mobile GPU coordination platform for augmented reality applications." Proceedings of the 26th Annual International Conference on Mobile Computing and Networking (2020).
[3] Romero, Francisco et al. "INFaaS: Automated Model-less Inference Serving." USENIX Annual Technical Conference (2021).
[4] Crankshaw, Daniel et al. "Clipper: A Low-Latency Online Prediction Serving System." NSDI (2017).

MicroEdge Vision

- A low-cost edge cluster serves computational resources for a set of geo-local cameras.
- A performant **multi-tenancy** architecture.



Objectives of MicroEdge System Architecture

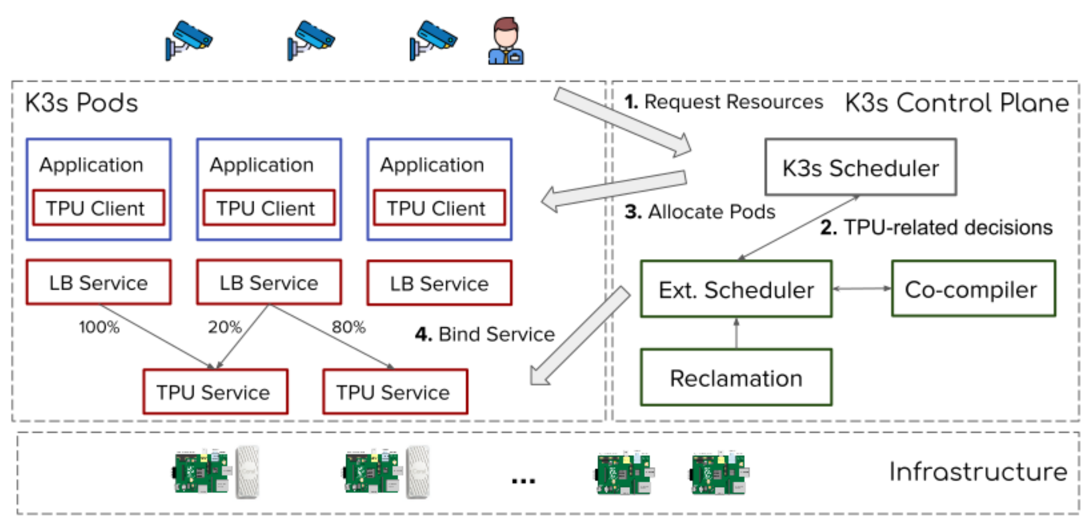
- Share TPU resources across independent application pipelines
- Performance isolation across independent application pipelines
- Load-balance incoming inference requests across TPU resources
- Co-compile different models on TPUs to reduce model swapping overhead

MicroEdge System Architecture

Research Questions
- **How do we extend the state of the art?**

Extensions to K3s's control plane
- Orchestrate TPU resources on creation and destruction of application pods
  - ➢ Admission Control
- Key components:
  - ➢ Extended scheduler, TPU co-complier, Resource reclamation

Extensions to K3s's data plane
- Mechanisms to multiplex TPU resources across applications
  - ➢ Monitoring TPU usage
- Key components:
  - ➢ TPU service, TPU service client library, Load balancing service



Evaluation Plan

Application Study

- Space-time vehicle tracking
- Can MicroEdge improve the resource utilization while meeting the SLA of application?

Trace-based Study

- Azure Function Traces [1]
- Map to ephemeral application pipelines
- Can MicroEdge effectively allocate and reclaim TPU resources?

[1] Shahrad, Mohammad, et al. "Serverless in the wild: Characterizing and optimizing the serverless workload at a large cloud provider." 2020 USENIX Annual Technical Conference (USENIX ATC 20). 2020.

Serverless at the Edge for camera networks

- Motivation: Workload at the edge is highly variable with time
  - Dedicating resources leads to overprovisioning
  - Scarce resources at the Edge
- Original vision for Coral-Pie was to create space-time track for all vehicles all the time
  - But most of the time, we are interested only in "suspicious" vehicles
  - Increases the chance for resource multiplexing
  - Number of interesting objects detected is sporadic

=> Move toward Serverless at the Edge

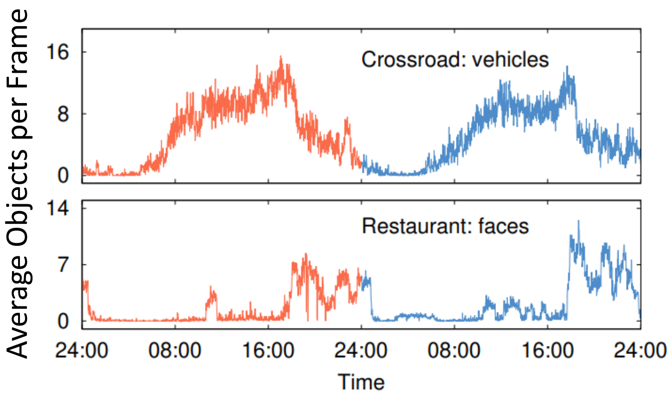
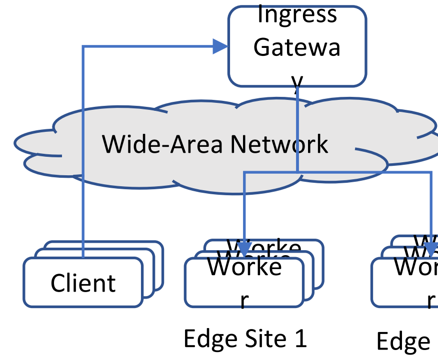
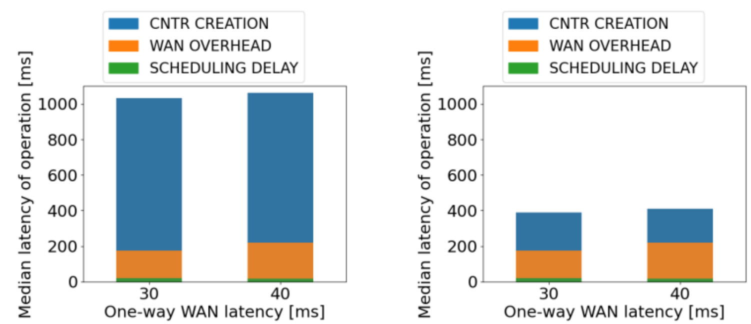


Image source: Zhang, Miao, et al. "Towards cloud-edge collaborative online video analytics with fine-grained serverless pipelines." *Proceedings of the 12th ACM Multimedia Systems Conference*. 2021.

Limitations of State-of-the-Art

- Cloud-based FaaS platforms
  - OpenFaaS, KNative (backed by K8s)
  - High Latency for OpenFaaS in MicroEdge
    ==> High container creation overhead
  - Multiple WAN traversals
    ==> High WAN overhead
  - Location-agnostic scheduling





(b) Latency breakdown with container cold start.

(c) Latency breakdown with pre-warmed containers.

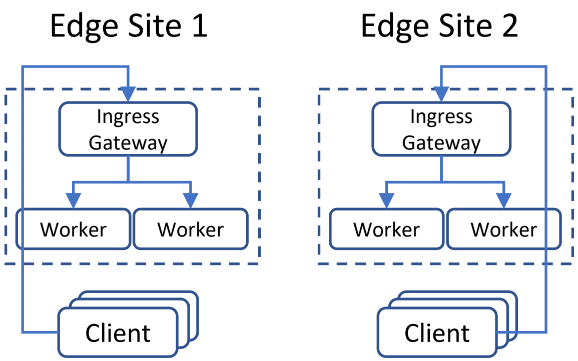
Figure 2: Experimental results with Kubernetes.

WAN latency = 80 ms RTT

Limitations of State-of-the-Art

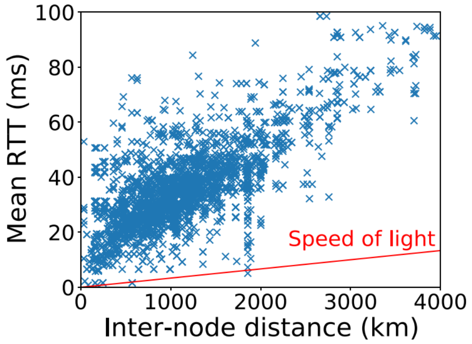
- Cloud-based FaaS platforms
  - OpenFaaS, KNative (backed by K8s)
  - High Latency for OpenFaaS in MicroEdge
  - Multiple WAN traversals
  - Location-agnostic scheduling

- Edge-specific FaaS platforms
  - Mu (SoCC'21)
    - Fine-grained queue-size based load-balancing to meet latency requirement
  - CEVAS (MMSys'21)
    - Optimal partitioning of serverless video processing between edge and cloud
  - Both platforms focus only on resources of 1 edge site
  - Load-balancing and Scaling policies are unaware of load on other sites



Objectives for Serverless @ Edge

- Cross-site load-balancing of function invocation requests
- Location-aware auto-scaling
- Nimble edge execution environment
- Handle heterogeneous network latencies between edge sites



Inter-edge-site latencies

Image source: Xu, Mengwei, et al. "From cloud to edge: a first look at public edge platforms." Proceedings of the 21st ACM Internet Measurement Conference. 2021.

Initial set of Research Questions

- Cross-site load balancing policy
  - Whether and to which site to offload?
- Global autoscaling policy
  - How to predict spatio-temporal distribution of client requests?
  - Where to provision function containers to minimize resource wastage and meet application requirement?
- Monitoring
  - What metrics are needed to enable above policies?
  - How to efficiently and scalably propagate per-site system state to other sites?
- Agility-oriented optimizations in Kubernetes to reduce cold start overhead
- Network proximity monitoring
  - How to do so accurately, efficiently, and at scale? (Network coordinates?)

Evaluation Plan

- Load balancing policy
  - Proposed network proximity and load-aware approach
  - Compare with
    - Coarse-grained monitoring (e.g., uniform load distribution *a la* OpenFaaS)
    - Fine-grained monitoring (e.g., fine-grained queue-sizes *a la* Mu)
- Autoscaling policy
  - Proposed spatio-temporal demand distribution based approach
  - Compare with
    - Techniques based on CPU utilization/request rate (*a la* OpenFaaS) and latency-aware (*a la* Mu)
- Network proximity
  - Is the use of network coordinates accurate, efficient, and scalable ?

Expected Contributions of the Proposed Research

MicroEdge

- Control and data-plane mechanisms for orchestrating accelerator resources
- Policies for efficient allocation of accelerator resources
- Admission Control and Monitoring policies

Serverless @ Edge

- Cross-site load and latency aware load-balancing policy
- Spatio-temporal demand based cross-site autoscaling policy
- Integrating network coordinates for network proximity monitoring

END