Processing Camera Streams Using Hierarchical Clusters

Chien-Chun Hung, Ganesh Ananthanarayanan, Peter Bodik,
Leana Golubchik, Minlan Yu, Victor Bahl, Matthai Philipose

Microsoft Research, University of Southern California (USC), Harvard

Cameras Are EVERYWHERE!

seattle.gov

Seattle Police Receive $600,000 Federal Grant For Body Cameras

THE WALL STREET JOURNAL.

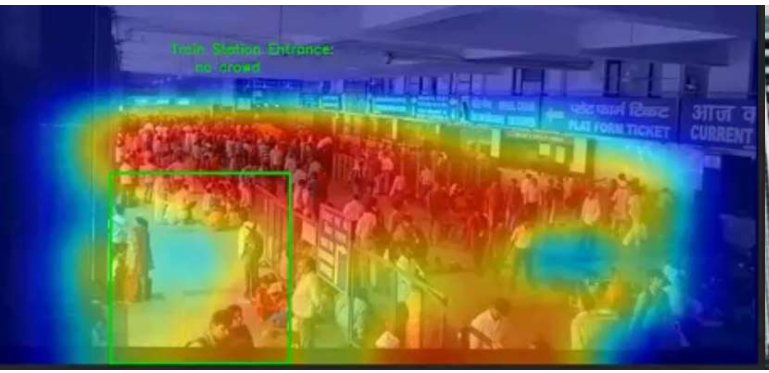
China's 100 Million Surveillance Cameras

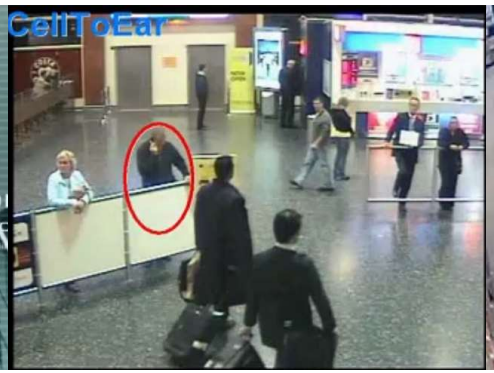
theguardian

You're being watched: there's one CCTV camera for every 32 people in UK

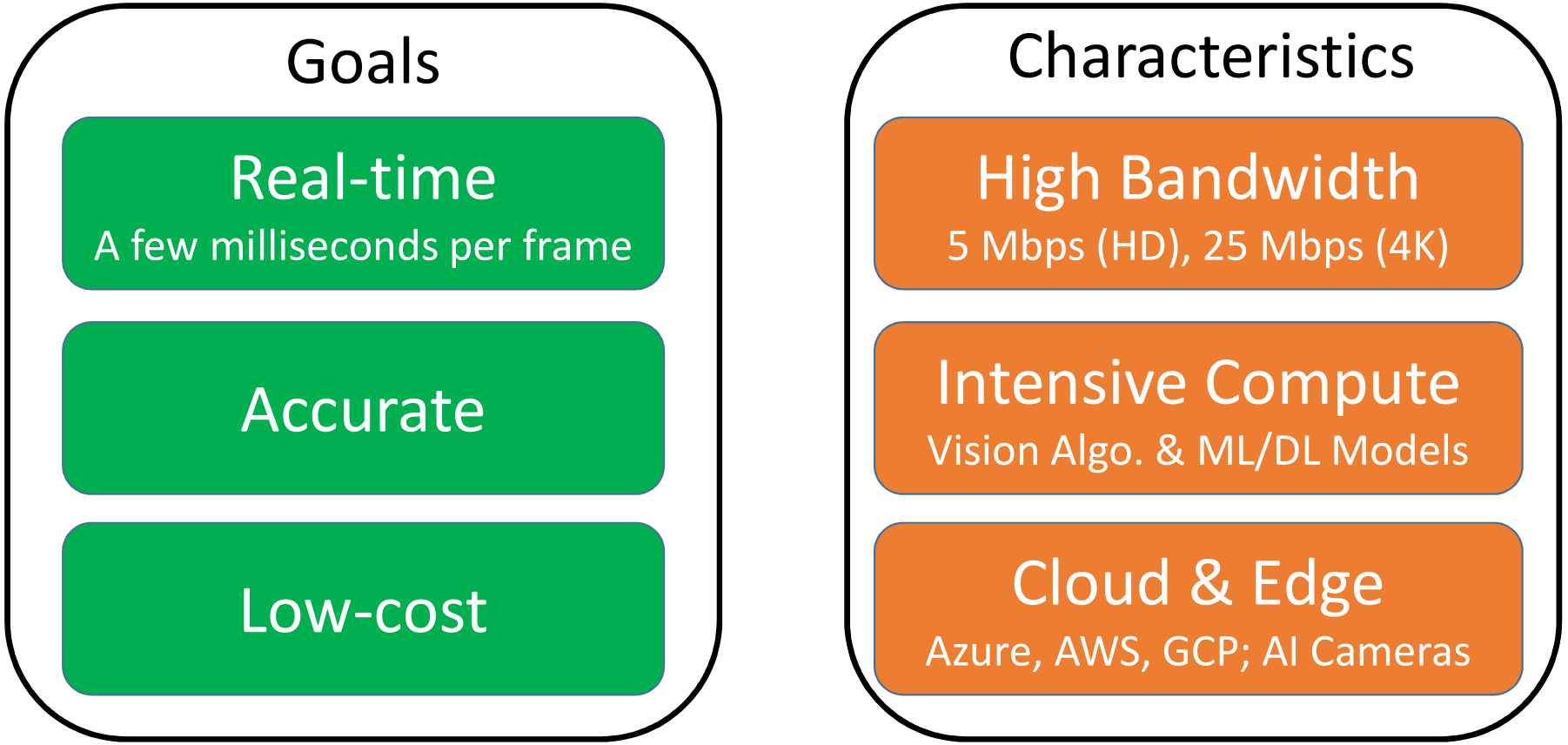
REUTERS

NYPD expands surveillance net to fight crime as well as terrorism



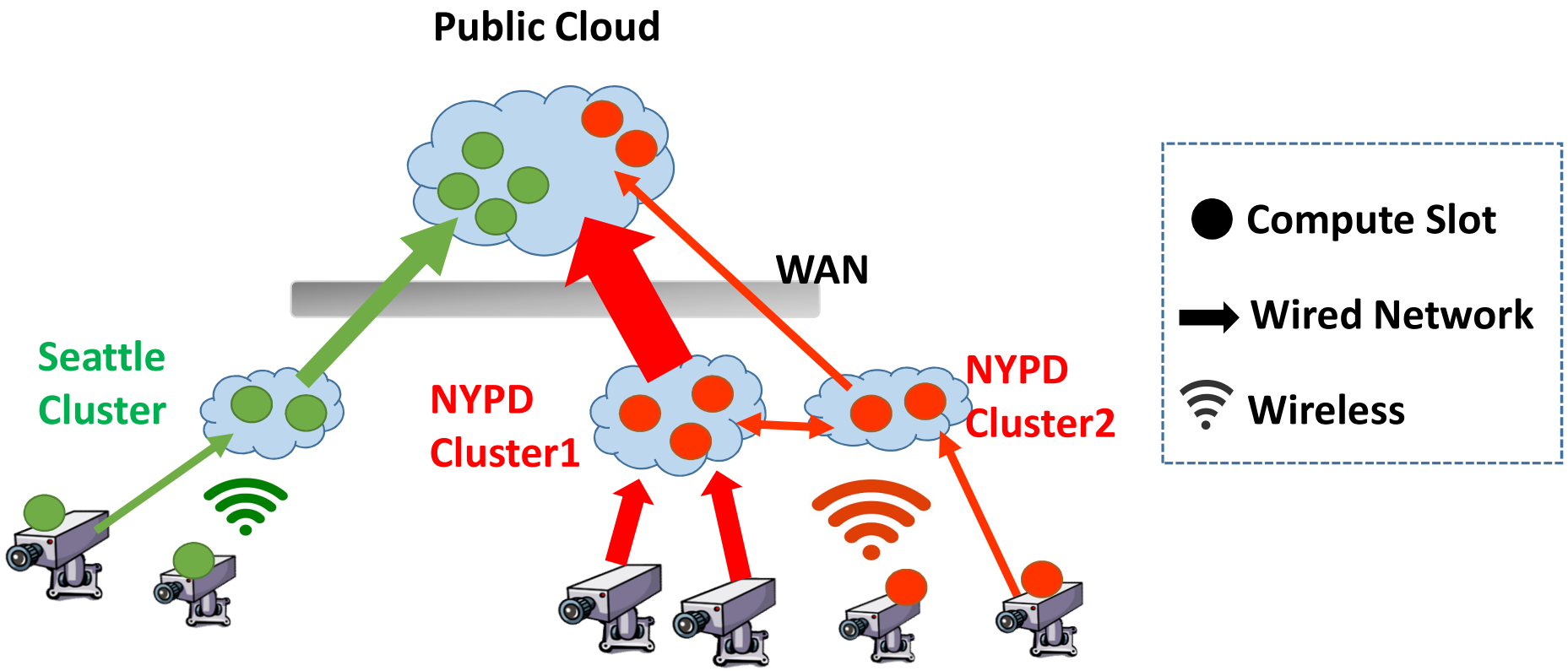


Video Analytics for Live Camera Streams

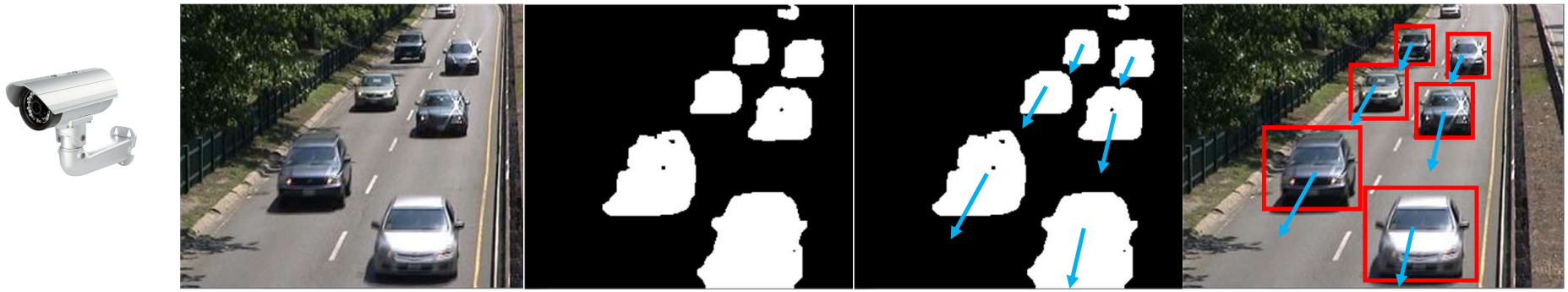


Hierarchical Computing Is The Key!

Hierarchical Clusters for Live Video Analytics



Video Query: Pipeline of Components



Video stream → **Decoder** → frame → **Detector** → object → **Associator** → track → **Application**

- ❖ Resolution
- ❖ Frame rate

40+ implementations

- ❖ Background subtraction (BGS)
- ❖ DNN-based
- ❖ Exhaustive search

60+ implementations

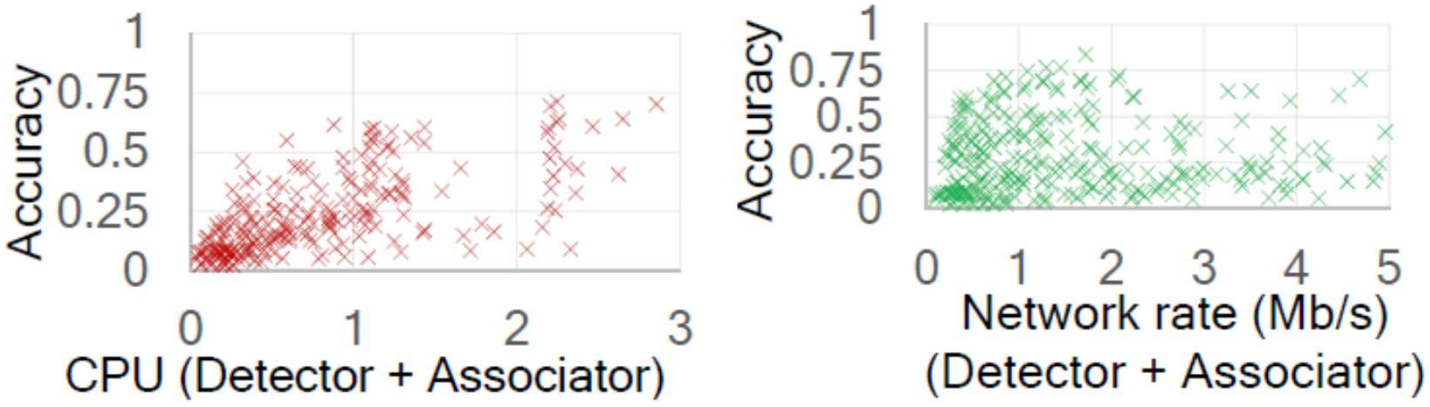
- ❖ Moving pattern (DIST)
- ❖ Color histogram (HIST)
- ❖ Key-point features (SURF, SIFT)

- ❖ Directional car counting
- ❖ Near-misses
- ❖ Anomaly detection

Many parameters and implementation options!

Video Query Configuration and Profiles

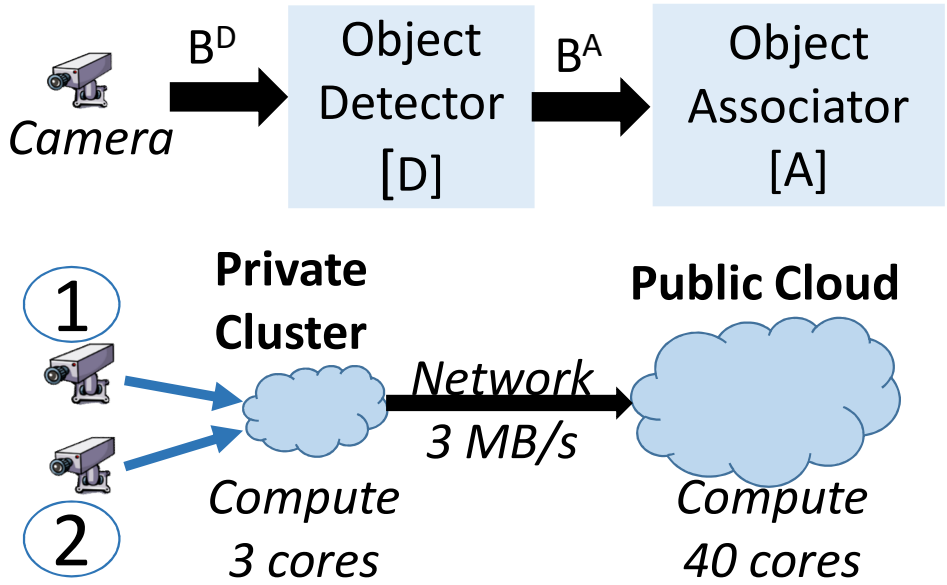
- **Planning**: select a combination of implementations/parameters for each component
- **Placement:** place query components across hierarchical clusters
- Each **configuration** (plan X placement) has a resource-accuracy profile



Thousands of **configurations** for each video query!

Illustrative Example
Planning and Placement

- 2 queries; 2 components each
- Q_{1080p} for both is infeasible
  - Query plans should be jointly determined across queries
- "Q_{1080p} + Q_{480p}" is the best, but only feasible under certain placement
  - Placement must be jointly considered with planning
- **Large decision space for joint planning and placement across multiple queries!**



Query Plan	B^D	B^A	C^D	C^A	Accuracy
Q_{1080p}	3	1.5	3	3	0.9
Q_{480p}	1	1	2	2	0.6
Q_{240p}	0.5	0.5	0.5	0.5	0.2

Real-time, Accurate and Low-cost Video Analytics

VideoEdge

Efficient Selection of Query Configuration to Maximize Accuracy

Evaluate The Cost of A Query Configuration

- Each query configuration has certain demand in each resource type
  - **CPU demand** in cameras, local clusters, public cloud
  - **Bandwidth demand** in the links in between

- Cost of a query configuration
  - **Dominant resource utilization** among all resource types

$$cost = \max_{resource:\, t} \frac{demand_t}{capacity_t}$$

- Avoid quickly draining out critical resources

Select The Query Configurations

- Greedy scheduling heuristic
  1. Start with baseline configurations
  2. Upgrade a configuration with maximum improvements
  3. Repeat 2. until ***resources deplete***, or ***no further upgrade can be done***

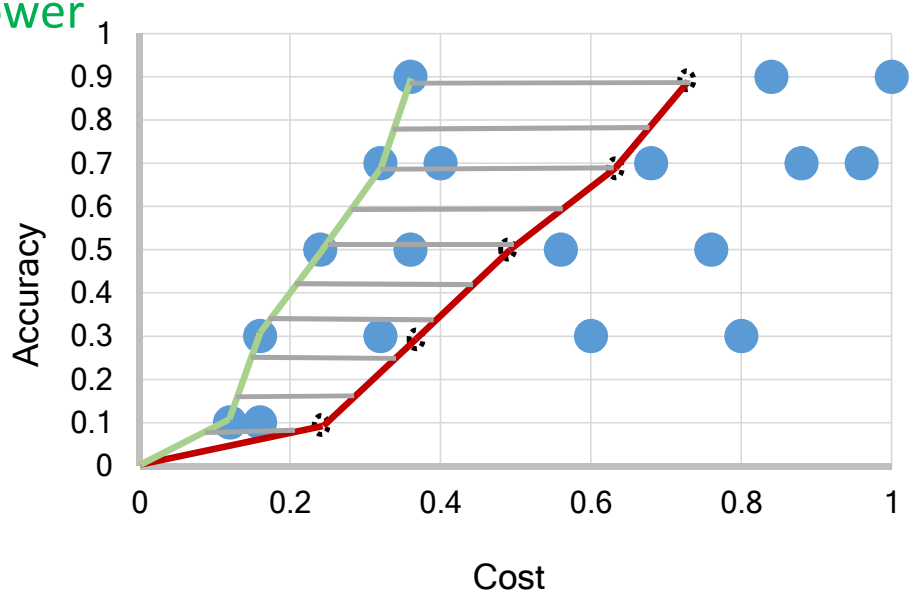
- Feasible configurations at any time
- Continuous improvements in accuracy
- Guarantee to converge with bounded iterations

Reduce Search Space with Pareto Band

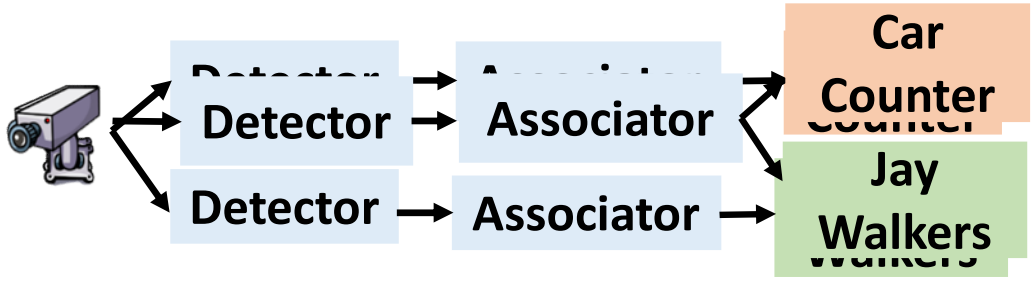
- Pre-filter out non-promising configurations

- Pareto boundary
  - No other point has higher accuracy ***AND*** lower cost than points on Pareto boundary

- Pareto band
  - More configurations to avoid infeasibility
  - Reduce running time by 75%



Improve Accuracy by Merging Queries



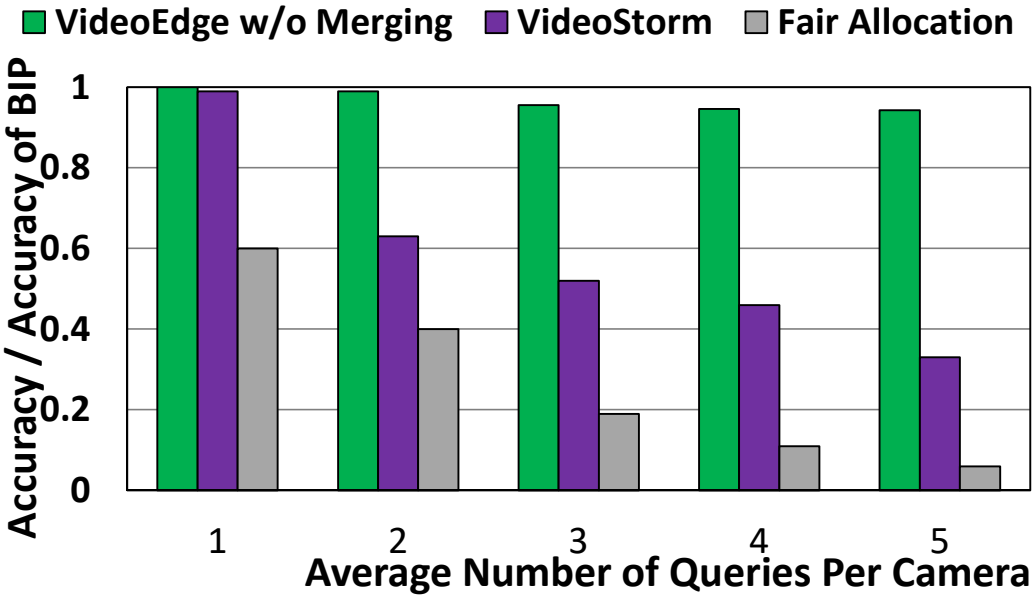
- Merging ***peer queries*** by running one-copy of common components
  - Save resources → improve overall accuracy
- Same configuration (plan & placement) for merged queries
  - Potential conflicts between the queries

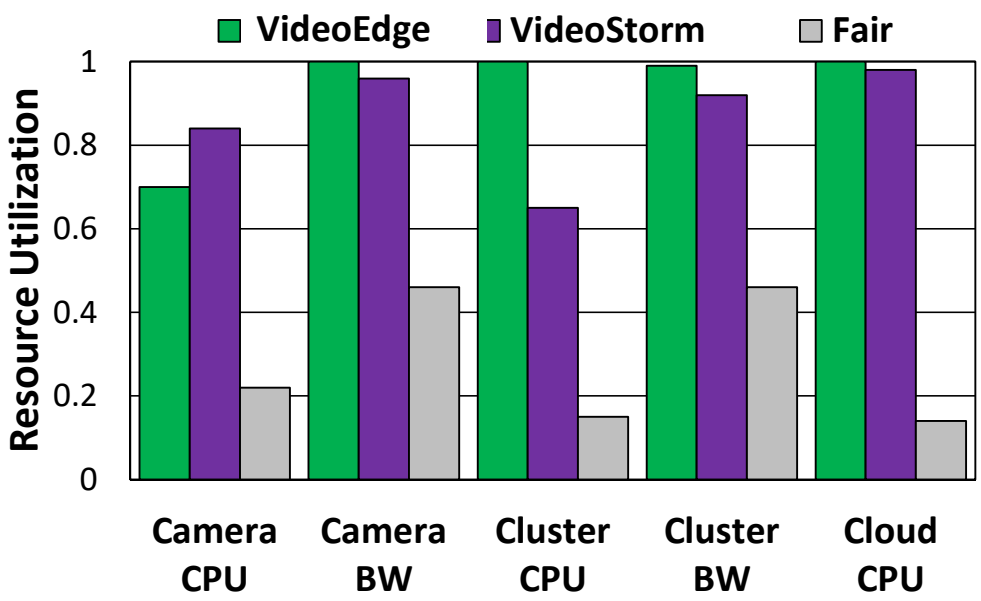
Evaluation Setup

- Azure deployment
  - A 25-node hierarchical cluster: 20 cameras, 2 private clusters and a cloud.
- Comparisons
  - **Optimal**: obtained by solving BIP optimization with Gurobi solver
  - **Fair Scheduler**: each query picks the best configuration within 1/n share
  - **VideoStorm**: schedule queries based on CPU resources only

Query Type	Num. Config.
Object Tracking	300
DNN Classifier (Object Classification)	20
License Plate Reader (Object Recognition)	30
Car Counting (Object Movement Stats)	10

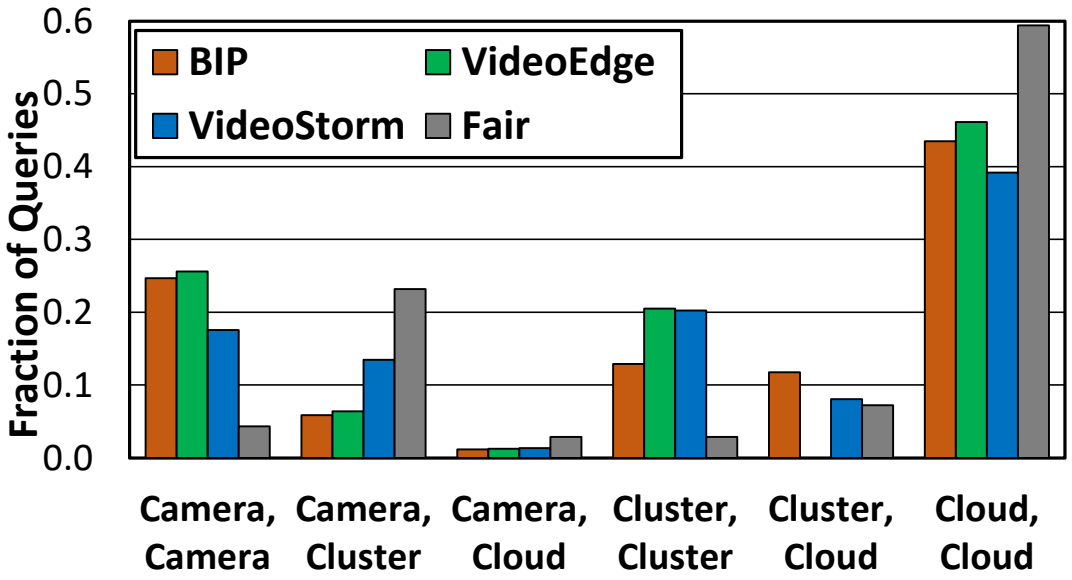
Improvement in Accuracy

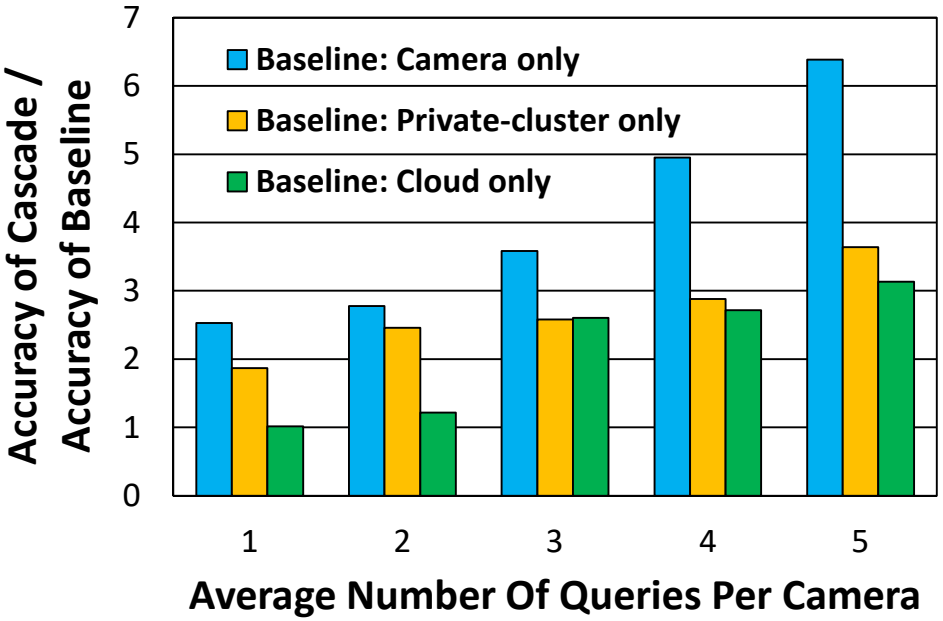




- VideoEdge achieves **94%** optimal even at high system load
  - **15.7X** better than fair scheduler, and **2.3X** better than VideoStorm
  - ***Near-optimal*** accuracy distribution
- VideoEdge achieves effective resource utilization
  - **70% (up)** utilization of all resource types

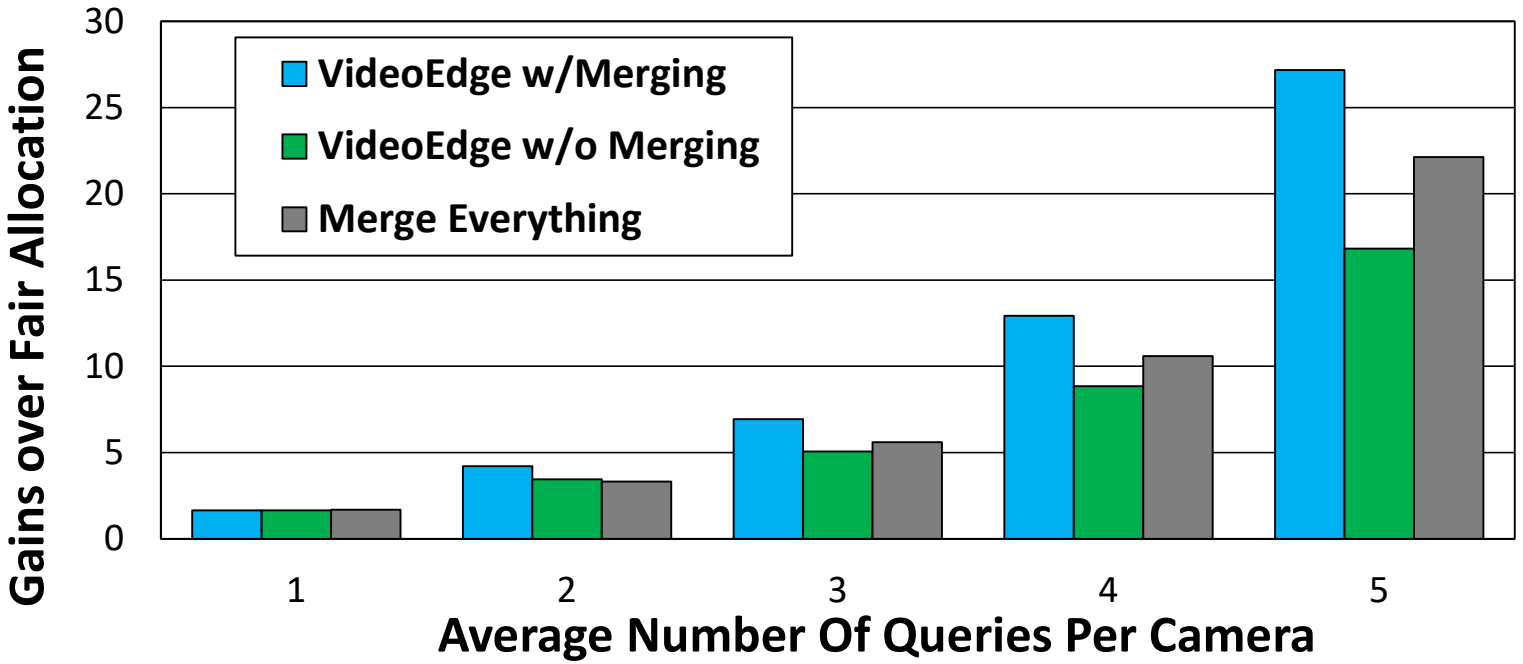
Gains with Placement Decisions





- VideoEdge places both components at the same location for **93%** queries
  - Saves inter-site bandwidth for other queries
  - Achieved by the cost metric (dominant resource utilization)
- VideoEdge is **3X** better compared to placement-restricted baselines

Gains with Merging Peer Queries



- Merging further provides **1.6X** gains for VideoEdge
  - **25.4X** better than Fair Scheduler, **5.4X** better than VideoStorm
- Blindly merging hurts overall accuracy even though it saves maximum resources

VideoEdge Conclusion

- ***Real-time***, ***accurate*** and ***low-cost*** video analytics for camera streams

- Contributions
  - ✓Define cost of a query configuration based on dominant resource utilization
  - ✓Efficiently schedule the queries with a greedy heuristic
  - ✓Further reduce search space with Pareto band
  - ✓Improve overall accuracy by merging queries

- Results
  - **25.4X** and **5.4X** better accuracy compared to fair scheduler and VideoStorm
  - Part of **Project Rocket** (http://aka.ms/rocket), deployed in Bellevue City