Live Video Analytics at Scale with Approximation and Delay-Tolerance

Haoyu Zhang, Ganesh Ananthanarayanan, Peter Bodik, Matthai Philipose, Paramvir Bahl, Michael J. Freedman
Video cameras are pervasive

NYPD expands surveillance not to fight crime and (video) art

Cameras and IoT: Going from smart to intelligent

Microsoft looks to stop bike crashes before they happen, testing Minority Report-style predictive intelligence
Video analytics queries

Intelligent Traffic System

AMBER Alert

Electronic Toll Collection

Video Doorbell
Video query: a pipeline of *transforms*

- Vision algorithms chained together
- Example: traffic counter pipeline
Video queries are expensive in resource usage

- Best car tracker \(^1\) — 1 fps on an 8-core CPU
- DNN for object classification \(^2\) — 30GFlops

When processing *thousands* of video streams in multi-tenant clusters
- How to reduce processing cost of a query?
- How to manage resources efficiently across queries?

\(^1\) VOT Challenge 2015 Results.
\(^2\) Simonyan et al. CVPR abs/1409.1556, 2014
Vision algorithms are intrinsically \textit{approximate}.

- **Knobs**: parameters / implementation choices for transforms

- License plate reader $\rightarrow$ window size
- Car tracker $\rightarrow$ mapping metric
- Object classifier $\rightarrow$ DNN model

- **Query configuration**: a combination of knob values
Knobs impact quality and resource usage

Frame Rate
- 3
- 1

Resolution
- 720p
- 480p

Quality
- 0.93, CPU=0.54
- 0.57, CPU=0.09
Knobs impact quality and resource usage

Frame Rate
Resolution
Window Size
Mapping Metric
Knobs impact quality and resource usage

- Orders of magnitude cheaper resource demand for little quality drop
- No analytical models to predict resource-quality tradeoff
  - Different from approximate SQL queries
Diverse quality and lag requirements

Lag: time difference between frame arrival and frame processing

- **Toll Collection**
  - Quality?: High
  - Lag?: Hours

- **Intelligent Traffic**
  - Quality?: Moderate
  - Lag?: Few Seconds

- **AMBER Alert**
  - Quality?: High
  - Lag?: Few Seconds
Goal

Decide configuration and resource allocation to maximize quality and minimize lag within the resource capacity.
Video analytics framework: Challenges

1. Many knobs → large configuration space
   • No known analytical models to predict quality and resource impact
2. Diverse requirements on quality and lag
   • Hard to configure and allocate resources jointly across queries
VideoStorm: Solution Overview

Profiler

Scheduler

query

resource-quality profile

utility function

offline

online

Workers
VideoStorm: Solution Overview

- **Profiler**
  - Builds model
  - Reduces config space

- **Scheduler**
  - Trades off quality and lag across queries

- **Utility Function**

- **Offline**

- **Online**
VideoStorm: Solution Overview

Profiler

Scheduler

Workers

query

resource-quality profile

utility function

offline

online
Offline: query profiling

- Profile: configuration $\Rightarrow$ resource, quality
  - Ground-truth: labeled dataset or results from *golden* configuration
  - Explore configuration space, compute average resource and quality

![Diagram showing resource demand vs. quality](image)

$\otimes$ is strictly better than $\otimes$ in quality and resource efficiency

higher quality

more efficient
Offline: Pareto boundary of configuration space

- **Pareto boundary**: optimal configurations in resource efficiency and quality
  - Cannot further increase one without reducing the other
  - Orders of magnitude reduction in config. search space for scheduling
VideoStorm: Solution Overview

query → Profiler → resource-quality profile → Scheduler → Workers

utility function

offline → online
Online: utility function and scheduling

- Utility function: encode \textit{goals} and \textit{sensitivities} of quality and lag
  - Users set required quality and tolerable lag
  - Reward additional quality, penalize higher lag

- Schedule for two natural goals:
  - \textbf{Maximize the minimum utility} – (max-min) fairness
  - \textbf{Maximize the total utility} – overall performance

- Allow lag accumulation during resource shortage, then catch up
Online: scheduling approximate video queries

- Queries: blue and orange (tolerate 8s lag)
- Total CPU: 4 → 2 → 4
- Fair scheduler: best configurations w/o lag
- Quality-aware scheduler: allow lag → catch up
Additional Enhancements

• Handle incorrect resource profiles
  • Profiled resource demand might not correspond to actual queries
  • Robust to errors in query profiles

• Query placement and migration
  • Better utilization, load balancing and lag spreading

• Hierarchical scheduling
  • Cluster and machine level scheduling
  • Better efficiency and scalability
VideoStorm Evaluation Setup

• Platform:
  • Microsoft Azure cluster
  • Each worker contains 4 cores of the 2.4GHz Intel Xeon processor and 14GB RAM

• Four types of vision queries:
  • license plate reader
  • car counter
  • DNN classifier
  • object tracker
Experiment Video Datasets

• Operational traffic cameras in Bellevue and Seattle

• 14 – 30 frames per second, 240P – 1080P resolution
Resource allocation during burst of queries

- Start with 300 queries:
  ① Lag Goal=300s, low-quality ~60%
  ② Lag Goal=20s, low-quality ~40%

- Burst of 150 seconds (50 – 200):
  ③ 200 LPR queries (AMBER Alert)
  High-Quality, Lag Goal=20s

- VideoStorm scheduler:
  ③ dominate resource allocation
  significantly delay ① run ② with lower quality
  All meet quality and lag goals
Resource allocation during burst of queries

• Start with 300 queries:
  ① Lag Goal=300s, low-quality ~60%
  ② Lag Goal=20s, low-quality ~40%

• Compare to a fair scheduler with varying burst duration:
  • Quality improvement: up to 80%
  • Lag reduction: up to 7x

• VideoStorm scheduler:
  significantly delay ① run ② with lower quality
  ③ dominate resource allocation
  All meet quality and lag goals
VideoStorm Scalability

- Frequently reschedule and reconfigure in reaction to changes of queries

- Even with thousands of queries, VideoStorm makes rescheduling decisions in just a few seconds
VideoStorm: account for errors in query profiles

- Errors in profile on resource demands
  - Over/under allocate resources → miss quality and lag goals!
- Example: 3 copies of same query, *should* get same allocation
  - Profiled resource synthetically doubled, halved and unchanged
- VideoStorm keeps track of mis-estimation factor $\mu$ – multiplicative error between the profiled demand and actual usage
Conclusion

- VideoStorm is a video analytics system that scales to processing thousands of video streams in large clusters

- **Offline profiler**: efficiently estimates resource-quality profiles
- **Online scheduler**: optimizes jointly for the quality and lag of queries

- VideoStorm is currently **deployed in Bellevue Traffic Department**, and soon will be deployed in more cities