Live Video Analytics at Scale with Approximation and Delay-Tolerance

Haoyu Zhang, Ganesh Ananthanarayanan, Peter Bodik, Matthai Philipose, Paramvir Bahl, Michael J. Freedman

Microsoft

PRINCETON UNIVERSITY

NSDI ’17
Video cameras are pervasive

- Major cities have millions of cameras deployed
  - in buildings: surveillance & business intelligence
  - on streets: traffic control & crime prevention

- Effective analysis of live video streams
  → key to achieve the potential of these cameras
Video analytics examples

Intelligent Traffic Systems

Emergency Systems

Electronic Toll Collection
Video query: pipeline of transformations

Chain of transformations
Resource usage for analytics

• Very expensive!
  • Car Tracker: 1 fps on an 8-core CPU
  • DNN for Object Classification: 30GFlops

• Processing thousands of video streams in multi-tenant clusters
  • How to reduce processing cost of a query?
  • How to manage resources efficiently across queries?
Parameters for vision algorithms

• Knobs: parameters / implementation choices for transforms

  • License plate reader → window size
  • Car tracker → mapping metric
  • Object classifier → DNN model

• Query configuration: combination of knob values
Configuration Impact on Resource Usage

- Frame Rate: 3, Resolution: 720p
  - Quality=0.93, CPU=0.54

- Frame Rate: 1, Resolution: 480p
  - Quality=0.57, CPU=0.09
Configuration Impact on Resource Usage

(a) License Plate — Resolution (sampling rate = 0.12)
(b) License Plate — Sampling (resolution = 480p)
(c) DNN — Sampling
(d) Tracker — Object Mapping
Quality and lag requirements

- Quality: measured as F1 score (harmonic mean between precision and recall)
- Lag: time difference between frame arrival and frame processing

<table>
<thead>
<tr>
<th></th>
<th>Toll Collection</th>
<th>Intelligent Traffic System</th>
<th>AMBER Alert</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality</td>
<td>High</td>
<td>Moderate</td>
<td>High</td>
</tr>
<tr>
<td>Lag</td>
<td>Hours</td>
<td>Few Seconds</td>
<td>Few Seconds</td>
</tr>
<tr>
<td>Pattern</td>
<td>On-going</td>
<td>On-going</td>
<td>Bursty</td>
</tr>
</tbody>
</table>
Goal

• Decide configuration and resource allocation to maximize quality and minimize lag within the resource capacity

• Challenges:
  • Many knobs → large configuration space
  • No known analytical model to predict quality and resource impact
  • Diverse requirements on quality and lag (e.g., Toll Collection vs. AMBER)
  • Hard to configure and allocate resources jointly across queries
VideoStorm Overview
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

Red configuration is strictly better than orange configuration in both quality and resource efficiency!
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 configuration search space for scheduling
Online: utility function and scheduling

• Utility function: encode goals and sensitivities of quality and lag
  • Users set required quality and tolerable lag
  • Reward additional quality
  • Penalize higher lag

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

• Allow lag accumulation during resource shortage, then catch-up
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
  • Predict the future query lag over a short time horizon

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

- Microsoft Azure cluster
- 1 manager machine for profiling and scheduling
- 100 worker machines
- Each worker contains 4 cores of 2.4GHz Intel Xeon processor and 14GB RAM
Evaluation Data

• Query types:
  • license plate reader
  • car counter
  • DNN classifier
  • object tracker

• Data:
  • 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
  • Lag goal=300s, low-quality (60%)
  • Lag goal=20s, low-quality (40%)

• 150s burst of 200 queries
  • Lag goal=20s, high-quality
  • e.g., AMBER alert

• Queries of burst delay queries of type 1, run type 2 with lower quality
Comparison to Fair Scheduler

- **Baseline:** work-conservative fair scheduler

- **VideoStorm outperforms baseline with:**
  - 80% in quality of outputs
  - 7x better lag
  - both resource allocation and query placement affect performance
Account for errors in profiles

- Difference between resource demand:
  - in resource-quality profile
  - actual demand (continuously monitored)

- Over/under allocate resources → miss quality and lag goals!
VideoStorm Scalability

- Frequently reschedule and reconfigure in reaction to changes of queries
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
- Currently deployed in Bellevue Traffic Department
Comments

• Valuable contribution
  • complete system
  • deployed in real-life scenario
  • significant improvements in performance
  • basis for other research

• Paper is relatively implementation and evaluation heavy
Questions
Precision and Recall

**Precision**
How many selected items are relevant?

**Recall**
How many relevant items are selected?

\[
\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}
\]

\[
\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}
\]
Utility function

\[ U(Q, L) = U^B + U^Q(Q) + U^L(L) \]
\[ = U^B + \alpha^Q \cdot (Q - Q^M)_+ - \alpha^L \cdot (L - L^M)_+ \]
Model-Predictive Control

- aim to improve performance in near future
- model performance over short time horizon $T$
- plug predicted lag into the utility function

\[
L_{k,t+T}(a_k, c_k) = L_{k,t} + T - T \frac{a_k}{D_k(c_k)}
\]