DATA ANALYTICS USING DEEP LEARNING

GT 8803 // FALL 2018 // JENNIFER MA

LECTURE #14: LIVE VIDEO ANALYTICS AT SCALE WITH APPROXIMATION AND DELAY-TOLERANCE

Georgia Tech

CREATING THE NEXT®

TODAY'S PAPER

- Live Video Analytics at Scale with Approximation and Delay-Tolerance

TODAY'S AGENDA

- Problem Overview
- Key Ideas
- Technical Details
- Experiments
- Discussion

PROBLEM OVERVIEW

- Querying camera recordings
- Traffic intersections, retail stores, offices, etc.
- Slow and costly

PROBLEM OVERVIEW

- Use cases?

PROBLEM OVERVIEW

- Use cases?
  - Catching criminals
    - Shoplifting
    - Trafficking
  - Sending ambulances
    - Car accidents
    - Free routes
  - Traffic control
  - Amber alerts

PROBLEM OVERVIEW

- 2 main problems with querying videos

PROBLEM OVERVIEW

- 2 main problems with querying videos
  - Slow
  - Costly

PROBLEM OVERVIEW

- Querying a month-long video would requires 280 GPU hours and $250
- To run the query in 1 minute requires 10000s of GPUs
- Traffic jurisdictions and retails may only have 10s or 100s
- VOT Challenge 2015 – 1 fps

PROBLEM OVERVIEW

- Goal: Optimize thousands of queries operating in clusters

KEY IDEAS

- 2 key characteristics of video analytics
  - Resource-quality tradeoff with multidimensional configurations
  - Variety in quality and lag goals

KEY IDEAS

- Resource-quality trade-off with multi-dimensional configurations

KEY IDEAS

- Resource-quality trade-off with multi-dimensional configurations
  - Estimated amount of resources needed
  - Quality: accuracy of output
  - Configuration: a combination of parameters for an algorithm
  - Multi-dimensional – how configurations have multiple parameters

KEY IDEAS

- Example parameters:
  - Video resolution
  - Frame rate
  - Size of the sliding window

KEY IDEAS

- Variety in quality and lag goals

KEY IDEAS

- Variety in quality and lag goals
  - Some outputs don't need to be 100% accurate, such as counts of cars
  - Some outputs can wait

KEY IDEAS

- Variety in quality and lag goals
  - Some outputs don't need to be 100% accurate, such as counts of cars
  - Some outputs can wait
    - Traffic tickets where the billing can be delayed

KEY IDEAS

- Variety in quality and lag goals
  - Some outputs don't need to be 100% accurate, such as counts of cars
  - Some outputs can wait
    - Traffic tickets where the billing can be delayed
  - Queries that need a fast result?

KEY IDEAS

- Variety in quality and lag goals
  - Some outputs don't need to be 100% accurate, such as counts of cars
  - Some outputs can wait
    - Traffic tickets where the billing can be delayed
  - Queries that need a fast result?
    - Amber alerts

KEY IDEAS

- Variety in quality and lag goals
  – Some outputs don't need to be 100% accurate, such as counts of cars
  – Some outputs can wait
    - Traffic tickets where the billing can be delayed
  – Queries that need a fast result?
    - Amber alerts
  – Outputs that need to have high accuracy?

KEY IDEAS

- Variety in quality and lag goals
  - Some outputs don't need to be 100% accurate, such as counts of cars
  - Some outputs can wait
    - Traffic tickets where the billing can be delayed
  - Queries that need a fast result?
    - Amber alerts
  - Outputs that need to have high accuracy?
    - Amber alerts

KEY IDEAS

- Variety in quality and lag goals
  - Some outputs don't need to be 100% accurate, such as counts of cars
  - Some outputs can wait
    - Traffic tickets where the billing can be delayed
  - Queries that need a fast result?
    - Amber alerts
  - Outputs that need to have high accuracy?
    - Amber alerts
  - Low accuracy?

KEY IDEAS

- Variety in quality and lag goals
  - Some outputs don't need to be 100% accurate, such as counts of cars
  - Some outputs can wait
    - Traffic tickets where the billing can be delayed
  - Queries that need a fast result?
    - Amber alerts
  - Outputs that need to have high accuracy?
    - Amber alerts
  - Low accuracy?
    - Counting cars

KEY IDEAS

- How do systems for stream processing allocate resources?

KEY IDEAS

- How do systems for stream processing allocate resources?
  - Resource fairness

KEY IDEAS

- How do systems for stream processing allocate resources?
  - Resource fairness

- VideoStorm, their system, takes into account the resource demand, the quality needed, and the lag tolerance. Lag is the amount of time that a frame has been waiting to be processed.

KEY IDEAS

- Challenges?

KEY IDEAS

- Challenges?
  - Hard to analyze what resources and the quality of the output needed for a query
  - Hard to pick configurations because there are many knobs
  - Trading off between lag and quality goals is tricky
  - Resource allocation across all queries each having many configurations is computationally intractable

KEY IDEAS

- Solution
  - Offline phase:
    - Analyze resource demand and quality needed of each query for different configurations
    - Pick the ones on the pareto boundary
  - Online phase:
    - Scheduler reallocates resources, reselects configurations, and considers migrating queries to different machines
    - Based on resource-quality profiles and changes in resource capacity

TECHNICAL DETAILS

Video queries specification:

- Queries are submitted to VideoStorm as sequences of transforms.
- A transform (task) could have multiple inputs and outputs

RESOURCE ALLOCATION

- Have a selection of configurations
- Pick configs for queries for overall better quality
- Put queries on lag if some queries with low lag-tolerance need resources

REAL-WORLD VIDEO QUERIES

- Examples

REAL-WORLD VIDEO QUERIES

- Examples
  – License plate reader
  – Car counter
  – Deep neural network classifier for object detection and classification
  – Object tracker

TECHNICAL DETAILS

– Parameters that affect CPU demand and quality for most video queries

TECHNICAL DETAILS

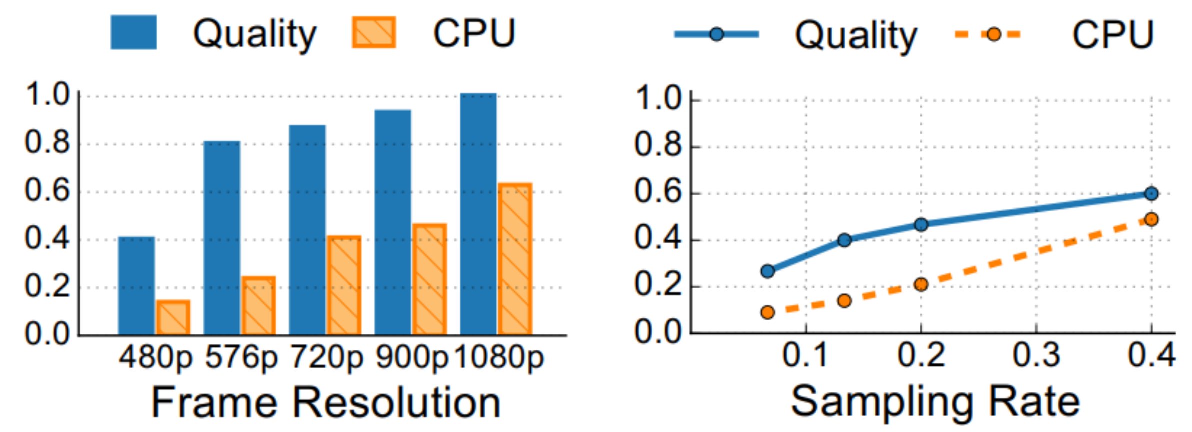
– Parameters that affect CPU demand and quality for most video queries
    - Image resolution
    - Frame sampling rate

TECHNICAL DETAILS

– How do these affect License plate reader queries?

TECHNICAL DETAILS

– How do these affect License plate reader queries?
  - Lower resolution and lower sampling rate lead to dramatically less resource demand
  - Missed or incorrectly read plates



(a) License Plate — Resolution (sampling rate = 0.12)

(b) License Plate — Sampling (resolution = 480p)

TECHNICAL DETAILS

- How do they affect a car counter?

TECHNICAL DETAILS

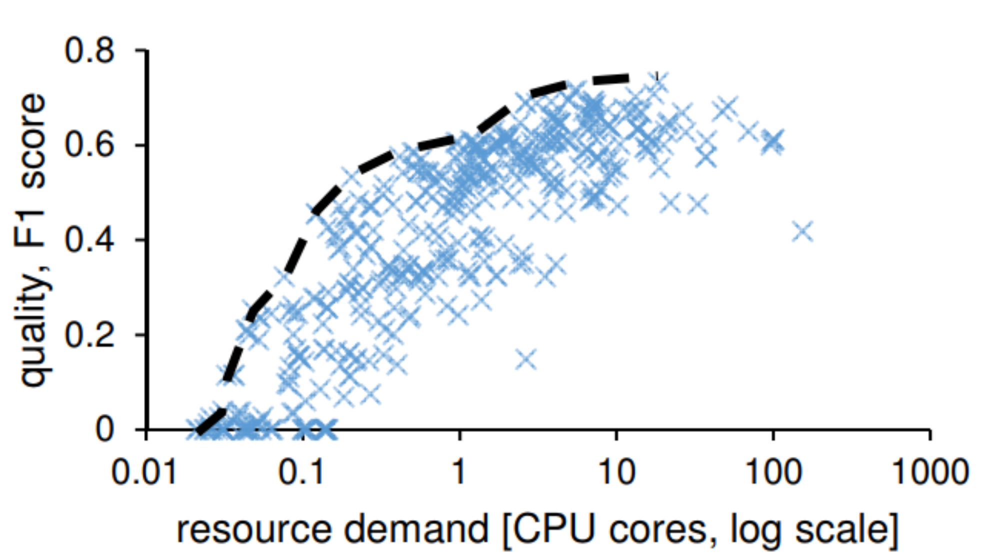
- How do they affect a car counter?
  - Good quality still

TECHNICAL DETAILS

- Profile estimation
  - Profile: estimated resources needed and desired accuracy of output
  - For a configuration of parameters, for one query

PROFILE ESTIMATION

– Overview
  - Pareto boundary



  - Compute a value for each profile

$$X(c) = Q(c) - \beta D(c)$$

PROFILE ESTIMATION

- Choosing configurations by greedy exploration
  - High quality and low demand
  - Hill climbing

TECHNICAL DETAILS

- Resource management:
  - Allocation – of resources for each query
  - Placement – of new and old queries

TECHNICAL DETAILS

- Utility function for a configuration
  - Quality and lag predicted
  - Utility is used to help select a configuration for a query

TECHNICAL DETAILS

Utility function:

Term	Description
$\mathcal{P}_k$	profile of query k
$c_k \in \mathcal{C}_k$	specific configuration of query k
$Q_k(c)$	quality under configuration c
$D_k(c)$	resource demand under configuration c
$L_{k,t}$	measured lag at time t
U_k	utility
Q_k^M	(min) quality goal
L_k^M	(max) lag goal
a_k	resources allocated

Table 2: Notations used, for query k.

$$U(Q,L) = U^B + U^Q(Q) + U^L(L)$$
$$= U^B + \alpha^Q \cdot (Q - Q^M)_+ - \alpha^L \cdot (L - L^M)_+$$

Baseline + bonus - penalty

TECHNICAL DETAILS

- Optimization objectives
  - Public cloud – maximize revenue -> maximize sum of utilities
  - Shared private cluster – want fairness -> maximize min utility

TECHNICAL DETAILS

- Resource allocation
  - Optimize for near future
  - Greedy approach

TECHNICAL DETAILS

Query placement

– Place new queries based on 3 goals
  - Maximizing utility in the cluster
  - Load balancing
  - Lag spreading

EVALUATION

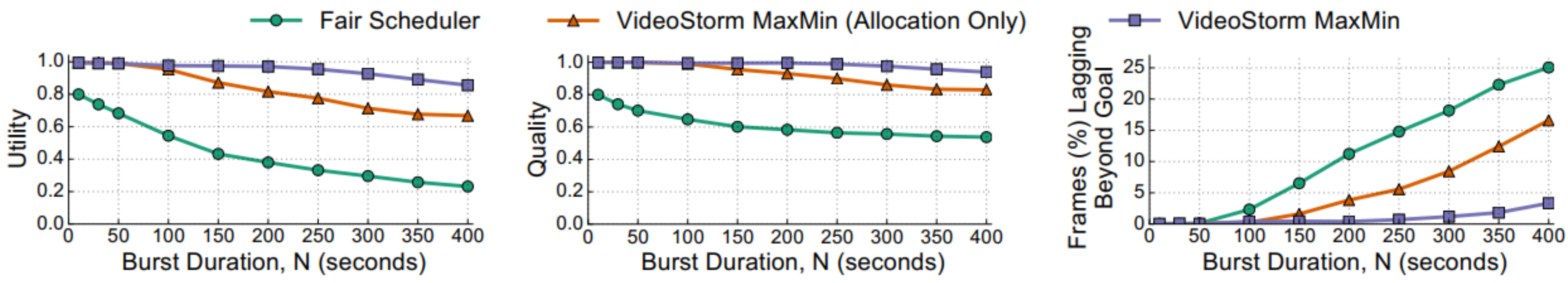
- Profiles are ‘nearly’ correct
- Setup
  - 4 types of queries
- Baseline
  - Fair scheduler
- Metrics
  - Quality
  - % frames exceeding lag goal
  - Utility

EVALUATION

- Performance
  - 300 queries of 4 types
  - Lag of 20s or 300s
  - Quality goal of 0.25
  - 300 ‘distinct’ video datasets

EVALUATION

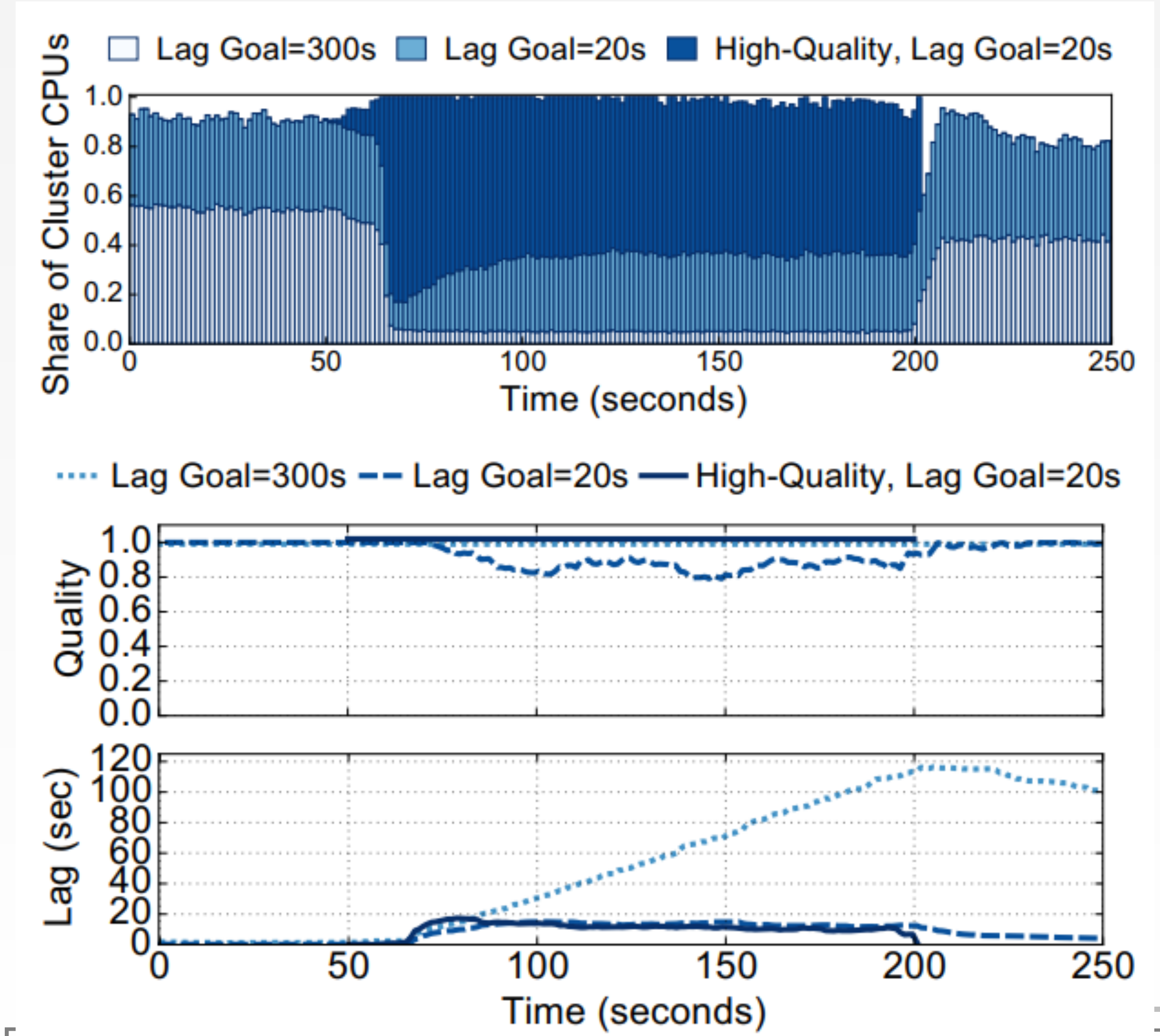
- Quality of fair scheduler(FS) is 0.2 lower to begin with
- Lowers to only 0.5 during a burst (200 license plate queries arrive)
- Quality for VideoStorm(VS) stays high at 90%
- Lag for FS keeps growing, VS stays low



(a) Utility (b) Quality (c) Lag

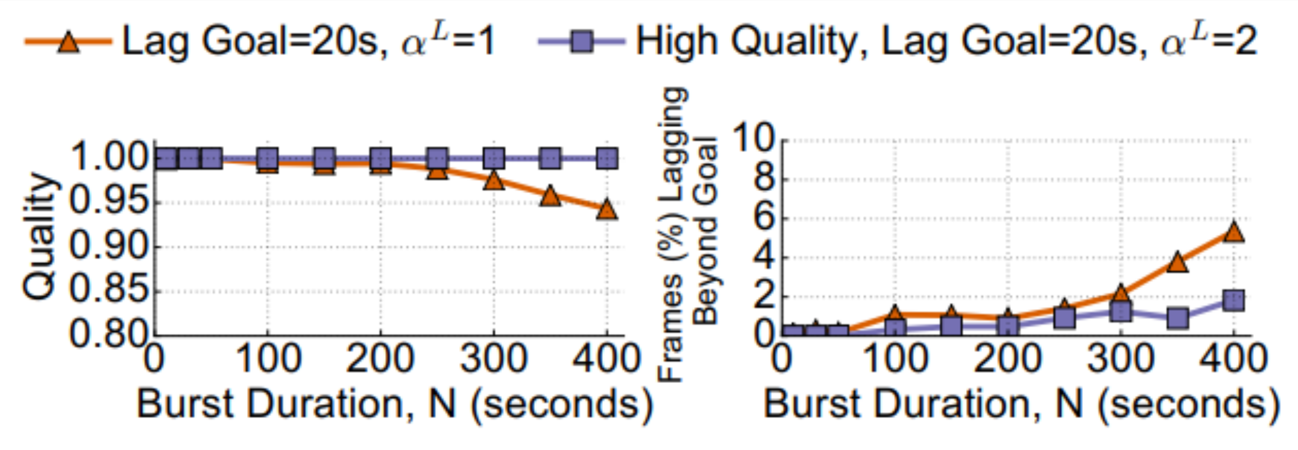
EVALUATION

- Burst in the middle
- More CPU's were allocated to queries with higher quality and short lag goal
- On the bottom, VS let lag accumulate only for queries with high tolerance



EVALUATION

- Can prioritize queries
- Using alpha
  - Higher alpha means higher priority
- In the graph, quality and lag is better for higher priority queries



STRENGTHS

- Used real VA queries, real traffic cameras, several cities
- Significant improvements: 80% increase in quality. 7x less lag
- Picks the knobs for the user
- Prioritizes queries
- Techniques are applicable to other stream analytics systems
- Gives bonus if a config has higher quality than the min, and punishes lag that is more than the max

WEAKNESSES

- Did not say if they add up the lag for each time step until T , or just at T.
- Did not talk about the approximation guarantees for the greedy algorithms
- Did not talk much about when profiles are wrong.
- Would have to tweak it to work with queries other than the 4 types

DISCUSSION

- Could it be combined into the ingestion part in Focus?
- Using machine learning to choose parameters
- Using machine learning to predict spikes, instead of the primitive formula for lag, so as to allocate more intelligently