ViCo: An Adaptive Distributed Video Correlation System

Xiaohui Gu, Zhen Wen, Ching-Yung Lin and Philip S. Yu
IBM T. J. Watson Research Center
On-line Data Stream Processing

High volume data streams (sensor data, financial data, media data)

Extracted result streams

real-time on-line processing functions/
Continuous query operator

Filter  Correlation  Clustering  Aggregation
Distributed On-line Video Correlation

Correlation predicate: $f(s_1, s_2)$
Sliding-windows:
W1 (e.g., 10 minutes)
W2 (e.g., 5 minutes)
Application Examples

- **Security surveillance**
  - Search suspicious scenes across surveillance videos
  - Correlate video streams captured by different camera sensors for cross validation

- **Information discovery**
  - Hot topic detection across different news videos

- **Workload reduction**
  - Filter out similar scenes for workload reduction
  - Avoid duplicate video processing
Overload Problem

- Resource-intensive and time-sensitive
  - Previous work: **Load Shedding**
    - Drop some frames predicted not important
    - Prediction could be wrong
    - Lose accuracy anyway
  - Our approach: **Load diffusion**
    - Scale-up without losing accuracy
    - Utilize distributed systems
Our Approach: Load Diffusion

Scale up stream processing without losing accuracy
Traditional Load Distribution

Load diffuser

S1 → 5 4 3 2 1 0
  s1 s1 s1 s1 s1 s1

S2 → 5 4 3 2 1 0
  s2 s2 s2 s2 s2 s2
Problem Formulation

Theorem: To distribute the workload of a stream correlation on multiple hosts, there must exist some video frames that are replicated on multiple hosts for preserving the correlation accuracy.

Problem definition: given a correlation operator and a set of hosts, each data item is distributed to one or more hosts such that

- correlation accuracy is preserved
- workload at different hosts is balanced
- diffusion overhead is minimized.
Our approach

- Route tuples to different hosts for join processing
  - Observe query semantics: windowed stream join
  - Adapt to dynamic stream environments
  - Fine-grained tuple-level load balancing
Single Stream Partition (SSP)

- Two stream roles
  - Master stream
  - Slave stream
- Split the master stream freely
- Replicate the slave stream Satisfy correlation constraint

Load diffuser
Algorithm Properties

Theorem: The single stream partition (SSP) algorithm preserves the query semantics: No missing join results or redundant join results.

Property: The overhead of the SSP algorithm is decided only by the rate of the slave stream.

\[(k - 1) \cdot r_2\]

Property: The diffusion efficiency of the SSP algorithm is decided only by the rate of the master stream.

\[\frac{r_1}{k} + r_2\]
Stream Role Adaptation

Adaptation strategy 1: dynamically select the master stream based on the arrival rates of input streams
Stream Role Switch Problem

Role switch

S₁ →

S₂ →

Load diffuser

s₁

s₂

s₁

s₁

s₁

s₂

s₂

s₂

s₂

s₂

s₂

s₂

s₂
On-line stream role switching

- Insert special marked video frames during the transition phase
  - * marked frames only join with usual frames before the switch time
  - # marked frames only join with usual frames after the switch time

The switch algorithm is proved to preserve join semantics
Coupled Stream Partition (CSP)

- Splits both streams
  - Split master stream into disjoint segments
  - Split slave stream into overlapped segments
  - Segment length $T$

Load diffuser
Theorem 1. The CSP algorithm preserves the accuracy of video stream correlations.

Property 1: The overhead of CSP is decided by the segment length and rate of the slave streams—**independent of the number of hosts**!

Property 2: The diffusion efficiency of CSP is decided by the rates of all input streams.

\[
\begin{align*}
    r_2 \cdot \frac{w_1 + w_2}{T} \\
    \sum_{i=1}^{n} \frac{r_i}{k} + \sum_{i=1, i \neq A}^{n} \frac{r_i \cdot w_A + w_i}{k \cdot T}
\end{align*}
\]
On-line segment length adaptation

- **Tradeoff** between overhead and balancing efficiency

- Theorem: The CSP segment change can preserve correlation accuracy when changing at the end of one segment

- Dynamic on-line profiling to tune the optimal segment length
On-line Stream role adaptation in CSP

- Dynamically select the master stream that can minimize algorithm overhead

Host set: \{ V_1, V_2, V_3, V_4 \}, W_2=2, W_1=3
Cross-algorithm adaptation

- SSP and CSP differs mainly in overhead
  - Overhead of SSP is related to the number of server nodes
  - Overhead of CSP is related to the sliding-window sizes
  - Theorem: the SSP algorithm has larger overhead than the CSP algorithm if
    \[
    k > \frac{w_1 + w_2}{T} + 1
    \]

  Adaptation strategy 4: dynamically switch between the SSP and CSP algorithms based on the algorithm overhead comparison.

- Semantics-preserving online algorithm switching
Experiment Evaluation

- **Distributed video correlation system**
  - Prototype implementation in C++
  - Running on real computer cluster
    - ~350 IBM blade servers

- **Workload**
  - Real video data stream
    - TRECVID-2005 data set
  - Video correlation specifications
    - Different number of streams
    - Different sliding-window sizes
    - Different number of operators
Prototype Results

Throughput for J1 = S1[60] >> S2[60]

Routing time for J1 = S1[60] >> S2[60]
Trace-driven simulation results
Conclusion

- Scale up video stream correlation while preserving accuracy
- First work on scalable on-line video correlation
  - A spectrum of semantics-preserving load diffusion algorithms
  - On-line semantics-preserving adaptations for adapting to dynamic stream environments

Future work
- Extend to multi-way stream correlations
- Cache intermediate join results to reduce redundant computation
- Integrate with other image analysis algorithms
- Multi-modal correlation
- Support other continuous query semantics