CAESAR: Context-Aware Event Stream Analytics in Real time

Olga Poppe, Chuan Lei, Elke A. Rundensteiner, and Dan Dougherty

March 18, 2016
Complex Event Processing

The same workload of independent event queries is continuously evaluated.
Application Context

- Event compositions signify application contexts
- Most event queries are appropriate only in certain contexts
- They can be safely suspended otherwise

Examples of application contexts:
- **Emergency management:** normal, crowded, fire
- **Health care:** safe, warning, violation
- **Algorithmic trading:** hold, buy, sell
- **Financial fraud:** approved, suspicious, fraud
Traffic Management Use Case

- **140 hours idling in traffic** due to congestion in 10-worst U.S. traffic corridors per year [The Wall Street Journal]
- **Health cost of $18 billion** due to traffic noise and pollution in the USA's 83 largest urban areas in 2010 [USA Today]
- **1.24 million deaths** due to traffic injuries worldwide in 2010 [Wikipedia]
Traffic Management Contexts

**Goal** is to leverage application contexts to speed up system responsiveness.
Challenges

• Rich semantics
  – Complex conditions implying a context
  – Unknown and unbounded context duration
  – Multiple inter-dependent event queries

• Readable specification

• Real time responsiveness
# State-of-the-art Approaches

<table>
<thead>
<tr>
<th></th>
<th>CEP Systems (Esper, StreamInsight)</th>
<th>CAESAR</th>
<th>Business Models (BPMN, UML)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expressive event queries</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Application contexts</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Context-aware optimizations</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
</tr>
</tbody>
</table>
Contributions & Outline

CAESAR system:
• Graphical model
• Context-aware algebra
• Context-driven optimization techniques
• Execution infrastructure

Performance evaluation
Outline

CAESAR Model
Context-aware Event Stream Analytics
Context-aware Event Stream Analytics

Application contexts
- clear
- congestion
- accident
- clear

Context deriving queries
- Many slow cars
- Stopped cars
- Few fast cars
- Traffic flows smoothly

Input event stream
- time
- $t_1$
- $t_2$
- $t_3$
- $t_4$

Position reports
Context-aware Event Stream Analytics

Output event stream

Context processing queries

Application contexts

Context deriving queries

Input event stream

Worcester Polytechnic Institute
Application Contexts

- Accident
- Clear
- Congestion

Diagram showing relationships between accident, clear, and congestion contexts.
Context Deriving Queries

- **Switch** if traffic flows smoothly
- **Switch** if stopped cars
- **Initiate** if many slow cars
- **Switch** if traffic flows smoothly
- **Initiate** if stopped cars
- **Initiate** if many slow cars

Diagram:

- Accident
- Clear
- Congestion
Context Processing Queries

- **accident**
  - **switch** if traffic flows smoothly
  - **initiate** if many slow cars
  - **terminate** if stopped cars removed

- **clear**
  - **switch** if traffic flows smoothly
  - **switch** if stopped cars
  - **initiate** if many slow cars
  - **terminate** if stopped cars

- **congestion**
  - **switch** if traffic flows smoothly
  - **initiate** if few fast cars
  - **terminate** if stopped cars removed

- **alarm computation**
- **toll computation**
Context-aware Event Queries

1. DERIVE NewCar(s.id, s.xway, s.dir, s.seg, s.lane, s.pos, s.sec)
   PATTERN SEQ(NOT Position f, Position s)
   WHERE f.sec+30 = s.sec AND f.id = s.id AND s.lane ≠ “exit”
   [CONTEXT congestion]

2. DERIVE Toll(c.id, c.sec, 5)
   PATTERN NewCar c
   [CONTEXT congestion]

3. INITIATE CONTEXT accident
   PATTERN Accident
   [CONTEXT congestion]

congestion → accident
Outline

CAESAR Algebra
Context-preserving Plan Generation

Human-readable visual model

INITIATE CONTEXT C0 Q0

C0

SWITCH CONTEXT Cf Q2

TERMINATE CONTEXT Cf Q3

Cf

Phase 1

Context deriving queries:

INITIATE CONTEXT C0 Q0

SWITCH CONTEXT Cf Q2 CONTEXT C0

TERMINATE CONTEXT Cf Q3 CONTEXT Cf

Context processing queries:

Q1 CONTEXT C0

Phase 2

Logical level: CAESAR model

Direct query plan

Machine-readable query set

Executable query plan

Physical level: CAESAR algebra
CAESAR Algebra Operators

1. Context initiation $CI_c(I, W)$
2. Context termination $CT_c(I, W)$
3. Context window $CW_c(I, W)$
4. Filter $FI_\theta(I)$
5. Projection $PR_{A,E}(I)$
6. Event pattern $P(I)$
Runtime Context Maintenance

Context bit vector $W$: $\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

Context types:
$c_a, c_b, ... c_z$

Time stamp $W.time$

- Updated by the context initiation & termination operators
- Accessed by the context window operator
- Synchronized by the time driven scheduler
**Translation from Query Set to Algebra Plan**

**DERIVE** Toll(c.id, c.sec, 5)

**PATTERN** NewCar c

**CONTEXT** congestion

**DERIVE** NewCar(s.id, s.xway, s.dir, s.seg, s.lane, s.pos, s.lane)

**PATTERN** SEQ(\(\text{NOT}\) Position f, Position s)

**WHERE**  
\[f.\text{sec} + 30 = s.\text{sec}\ \text{AND}\  
\ f.\text{id} = s.\text{id}\ \text{AND}\  
\ f.\text{lane} \neq \text{'exit'}\]

**CONTEXT** congestion

- **Projection:** c.id, c.sec, 5
- **Context window:** congestion
- **Pattern:** NewCar c
- **Projection:** s.id, s.xway, s.dir, s.seg, s.lane, s.pos, s.sec
- **Context window:** congestion
- **Filter:**  
  \[f.\text{sec} + 30 = s.\text{sec} \land \  
  f.\text{id} = s.\text{id} \land s.\text{lane} \neq \text{'exit'}\]

- **Pattern:** SEQ(\(\text{NOT}\) Position f, Position s)

**Event stream**
Outline

CAESAR Optimizer
Problem statement:

Given a workload of context-aware event queries, our optimization problem is to find an optimized query plan for this workload with minimal CPU cost.

Context-aware optimization techniques:

• Context window push down strategy
• Context workload sharing algorithm
Context Window Push Down Strategy

Performance benefits:
- Suspension of irrelevant operators
- Context-driven stream routing
Context Workload Sharing Algorithm

Initiate $c_1$ if $X > 10$

Initiate $c_2$ if $X > 20$

$w_{c_1}$ with queries $Q_1$, $Q_3$

Terminate $c_1$ if $X < 30$

Terminate $c_2$ if $X < 40$

$w_{c_2}$ with queries $Q_1$, $Q_2$
Context Workload Sharing Algorithm

- **Initiate** $c_1$ if $X > 10$
- **Terminate** $c_1$ if $X < 30$
- **Initiate** $c_2$ if $X > 20$
- **Terminate** $c_2$ if $X < 40$

- $w_{c1}$ with queries $Q_1$, $Q_3$
- $w_{c2}$ with queries $Q_1$, $Q_2$
- $w_{c11}$ with $Q_1$, $Q_3$
- $w_{c12}$ with $Q_1$, $Q_3$
- $w_{c21}$ with $Q_1$, $Q_2$
- $w_{c22}$ with $Q_1$, $Q_2$
Context Workload Sharing Algorithm

\[ w_{c1} \text{ with queries Q1, Q3} \]

\[ w_{c2} \text{ with queries Q1, Q2} \]

\[ w_{c11} \text{ with Q1, Q3} \]

\[ w_{c12} \text{ with Q1, Q3} \]

\[ w_{c21} \text{ with Q1, Q2} \]

\[ w_{c22} \text{ with Q1, Q2} \]

\[ w_{c11} \text{ with Q1, Q3} \]

\[ w_{c} \text{ with Q1, Q2, Q3} \]

\[ w_{c22} \text{ with Q1, Q2} \]
CAESAR Infrastructure & Experiments
CAESAR Architecture

- **Specification layer**: Visual context editor, Context-aware translator
- **Optimization layer**: Context-driven optimizer, Cost model
- **Execution layer**: Context-aware stream router, Scheduler, Context derivation, Context processing
- **Storage layer**: Context windows, Context history

Input event stream → Output event stream
Experimental Setup

**Execution infrastructure:**
Java 7, 1 Linux machine with 16-core
3.4 GHz CPU and 48GB of RAM

**Data sets:**
- Linear Road stream benchmark (LR) [1]
  3 roads=1.7GB
- Physical Activity Monitoring real data set (PAM) [2]
  1.6GB

[1] A.Arasu et al., Linear Road: A stream data management benchmark. VLDB’04
[2] A.Reiss et al., Creating and benchmarking a new data set for physical activity monitoring. PETRA’12
For 7 roads, context-aware (CA) event stream analytics is 9-fold faster than context-independent (CI) approach.
Context-aware Event Query Sharing

If 30 context windows of length 15 minutes process 4 event queries each and overlap by 15 minutes, workload sharing wins 6-fold.
Conclusions
Conclusions

- CAESAR is first context-aware CEP system
- Graphical context-specification model
- Context-aware algebra
- Context-driven optimization techniques
- Execution infrastructure
- 8-fold speed up on average
Acknowledgement

- Advisors: Elke A. Rundensteiner, Dan Dougherty
- Collaborator: Chuan Lei
- DSRG group at WPI
- EDBT reviewers
- NSF grants IIS 1018443 and IIS 1343620