Scalable Pattern Sharing over Event Streams

Medhabi Ray (Microsoft)
Chuan Lei (NEC Labs America)*
Elke A. Rundensteiner (WPI)

06/28/2016

Motivation – CEP

Event Producers

Event Streams

Event Processing Engine

Event Patterns

Event Pattern Consumers

Motivation – Multiple CEP

Traffic Event Stream

Observation

Multiple Pattern queries share ***common sub-patterns*** and are often ***run concurrently***.

Multiple Pattern Queries

Q1: SEQ (CarLoc c (c.pos = tollA),
CarLoc c1 (c1.pos = WorkZone, c1.id = c.id),
CarLoc c2 (c2.pos = exitA & c.id = c2.id))

Q2: SEQ (CarLoc c (c.pos = tollA),
Accident a (a.pos = tollA + 5 miles),
CarLoc c1 (c1.pos = exitA & c.id = c1.id))

Q3: SEQ (CarLoc c (c.pos = tollA),
CarLoc c1 (c1.pos = c.pos &
c1.type = EMS & c1.state = Parked),
CarLoc c2 (c2.pos = exitA & c.id = c2.id)

Q4: SEQ (CarLoc c (c.pos = tollA),
CarLoc c1 (c1.pos = WorkZone, c1.id = c.id),
CarLoc c2 (c2.pos = exitB & c.id = c2.id))

Challenges – Time-Ordered Sequence Correlation


```
Q1: SEQ (CarLoc c (c.pos = tollA),
     CarLoc c1 (c1.pos = WorkZone, c1.id=c.id),
     CarLoc c2 (c2.pos = exitA & c.id=c2.id))
```

```
Q2: SEQ (CarLoc c (c.pos = tollA),
     Accident a (a.pos = tollA+5 miles),
     CarLoc c1 (c1.pos = exitA & c.id=c1.id))
```

```
Q3: SEQ (CarLoc c (c.pos = tollA),
     CarLoc c1 (c1.pos = c.pos & c1.type= EMS & c1.state = Parked),
     CarLoc c2 (c2.pos = exitA & c.id=c2.id)
```

```
Q4: SEQ (CarLoc c (c.pos = tollA),
     CarLoc c1 (c1.pos = WorkZone, c1.id=c.id),
     CarLoc c2 (c2.pos = exitB & c.id=c2.id))
```

T_1 = 4:00 PM T_2 = 4:10 PM T_3 = 4:15 PM

Challenges – Inter-Query Event Correlation


```
Q1: SEQ (CarLoc c (c.pos = tollA),
     CarLoc c1 (c1.pos = WorkZone, c1.id=c.id),
     CarLoc c2 (c2.pos = exitA & c.id=c2.id))
```

```
Q2: SEQ (CarLoc c (c.pos = tollA),
     Accident a (a.pos = tollA+5 miles),
     CarLoc c1 (c1.pos = exitA & c.id=c1.id))
```

```
Q3: SEQ (CarLoc c (c.pos = tollA),
     CarLoc c1 (c1.pos = c.pos & c1.type= EMS & c1.state = Parked),
     CarLoc c2 (c2.pos = exitA & c.id=c2.id)
```

```
Q4: SEQ (CarLoc c (c.pos = tollA),
     CarLoc c1 (c1.pos = WorkZone, c1.id=c.id),
     CarLoc c2 (c2.pos = exitB & c.id=c2.id))
```

Not likely to occur TOGETHER

Challenges – Inter-Query Event Correlation


```
Q1: SEQ (CarLoc c (c.pos = tollA),
     CarLoc c1 (c1.pos = WorkZone, c1.id=c.id),
     CarLoc c2 (c2.pos = exitA & c.id=c2.id))
```

```
Q2: SEQ (CarLoc c (c.pos = tollA),
     Accident a (a.pos = tollA+5 miles),
     CarLoc c1 (c1.pos = exitA & c.id=c1.id))
```

```
Q3: SEQ (CarLoc c (c.pos = tollA),
     CarLoc c1 (c1.pos = c.pos & c1.type= EMS & c1.state = Parked),
     CarLoc c2 (c2.pos = exitA & c.id=c2.id)
```

```
Q4: SEQ (CarLoc c (c.pos = tollA),
     CarLoc c1 (c1.pos = WorkZone, c1.id=c.id),
     CarLoc c2 (c2.pos = exitB & c.id=c2.id))
```

Very likely to occur TOGETHER

Challenges – Intractable Search Space

Q1: SEQ (A, B, C, D, E)

Q2: SEQ (A, B, M, D, P)

Q3: SEQ (X, B, M, N, Y)

Q4: SEQ (X, B, C, D, P)

Shared Plan

Q1: SEQ (A, B, C, D, E)

Q2: SEQ (A, B, M, D, P)

Q3: SEQ (X, B, M, N, Y)

Q4: SEQ (X, B, C, D, P)

Shared Plan

Q1: SEQ (A, B, C, D, E)

Q2: SEQ (A, B, M, D, P)

Q3: SEQ (X, B, M, N, Y)

Q4: SEQ (X, B, C, D, P)

Shared Plan

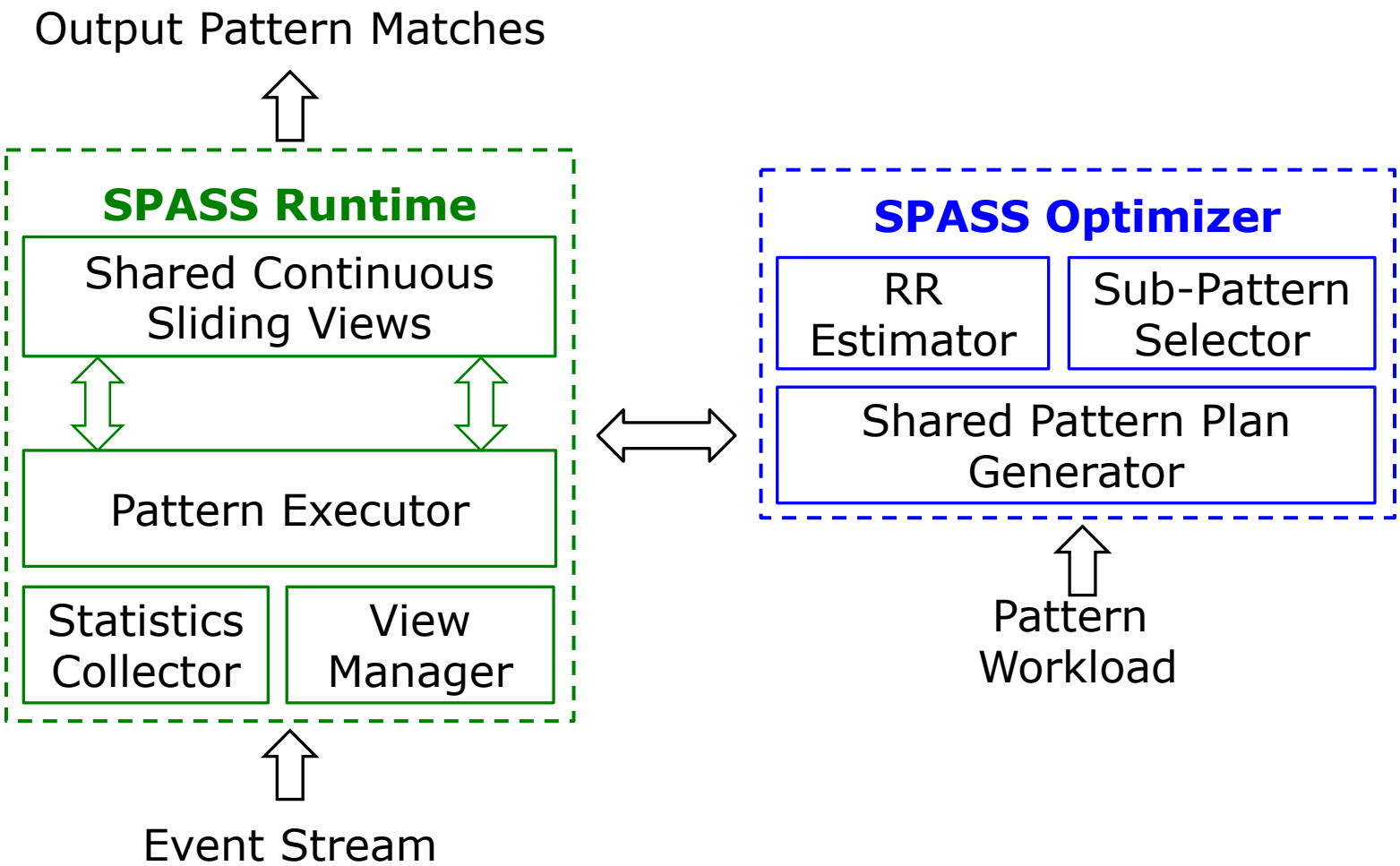
Q1: SEQ (A, B, C, D, E)

Q2: SEQ (A, B, M, D, P)

Q3: SEQ (X, B, M, N, Y)

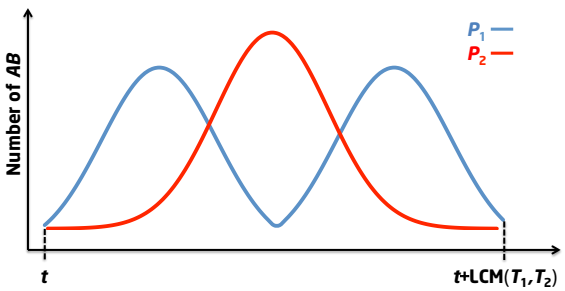
Q4: SEQ (X, B, C, D, P)

SPASS Architecture



SPASS Optimizer

- Intra-query event correlation
  - Number of event instances of E_j follow an event of E_i
  - Estimate # sub-pattern matches formed in a time interval
- Inter-query event correlation
  - Ratio between # of matches computed with sharing and # of matches computed separately
  - Estimate the degree of sharing across multiple patterns
- Redundancy Ratio (RR)
  - Estimate the degree of the redundant computation of matches using both Intra- and Inter-Query Event Correlations



SPASS Optimizer cont.

Workload

Q1: SEQ(A,B,C)

Q2: SEQ(B,C,D)

Q3: SEQ(A,B,C,D)

Q4: SEQ(B,C)

Identify shared sub-patterns

Query Map

Q1: <A, BC>

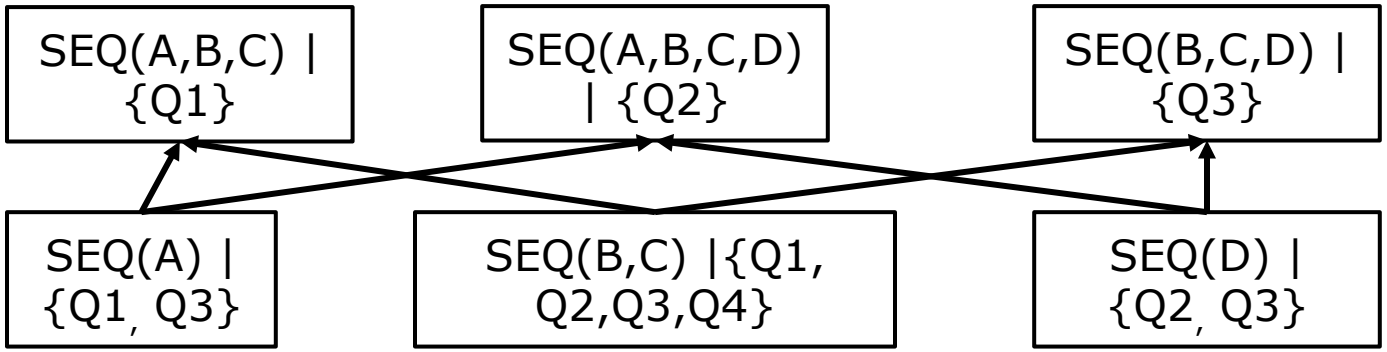
Q2: <BC, D>

Q3: <A, BC, D>

Q4: <BC>

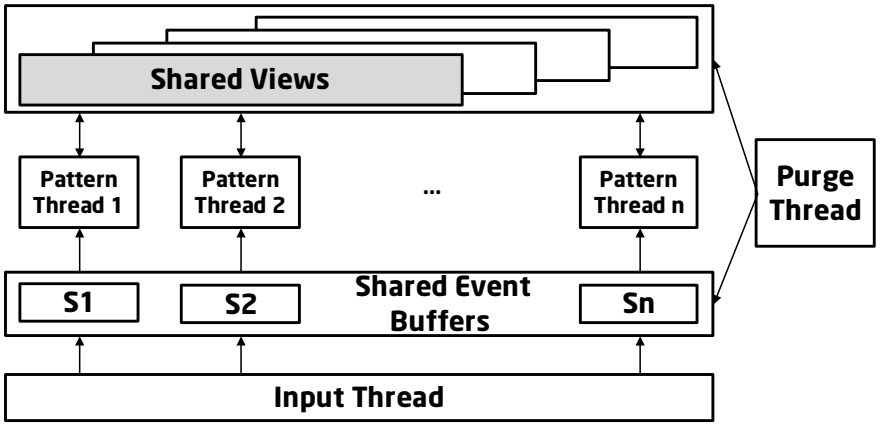
Build shared pattern plan

Shared Pattern Plan



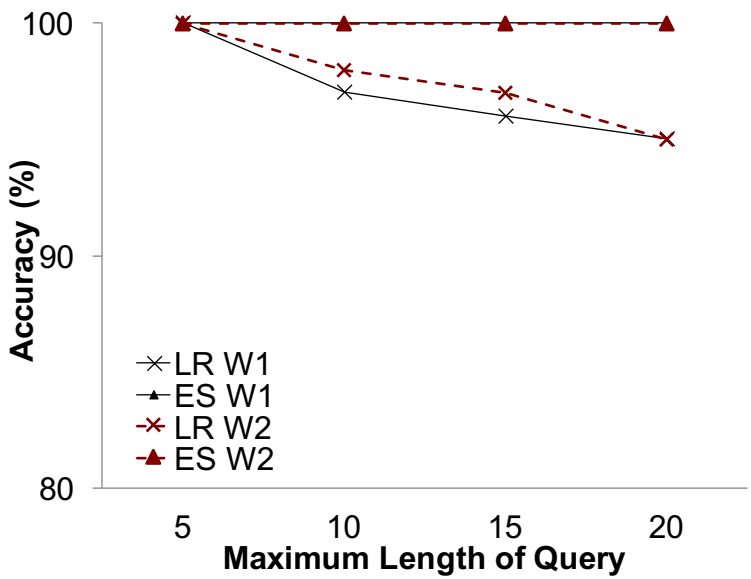
SPASS Runtime

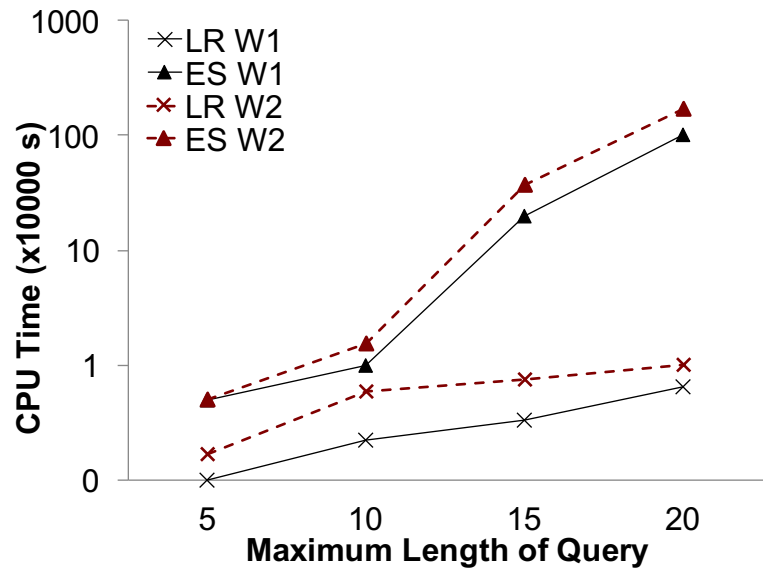
- Challenge – concurrent access to shared sub-pattern matches
- Solution – shared continuous sliding views store intermediate results of sub-patterns for subsequent reuse
  - View Validity Interval (VVI) timestamp-based indicators associated with materialized views
  - View Lookup Interval (VLI) a time interval to look up pattern matches



Experimental Results – Optimizer

- SPASS optimizer efficiently produces a nearly-optimal solution
  - Length of queries (5 – 20 event types)

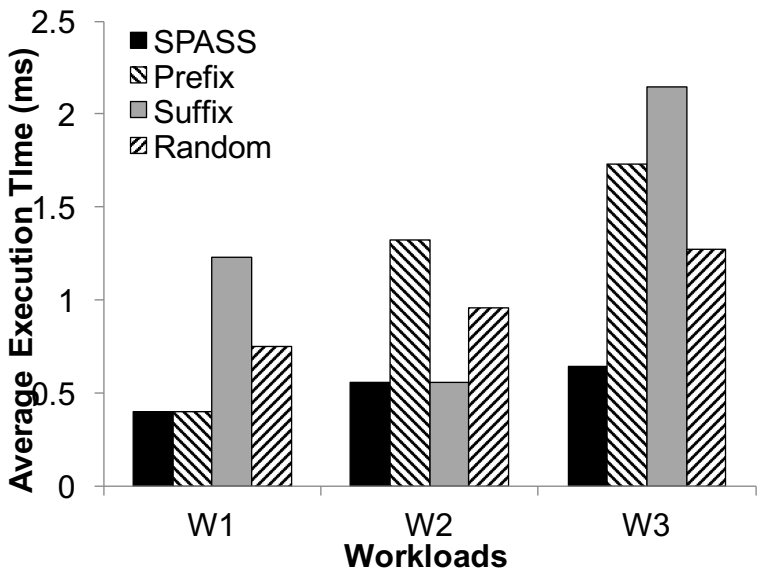


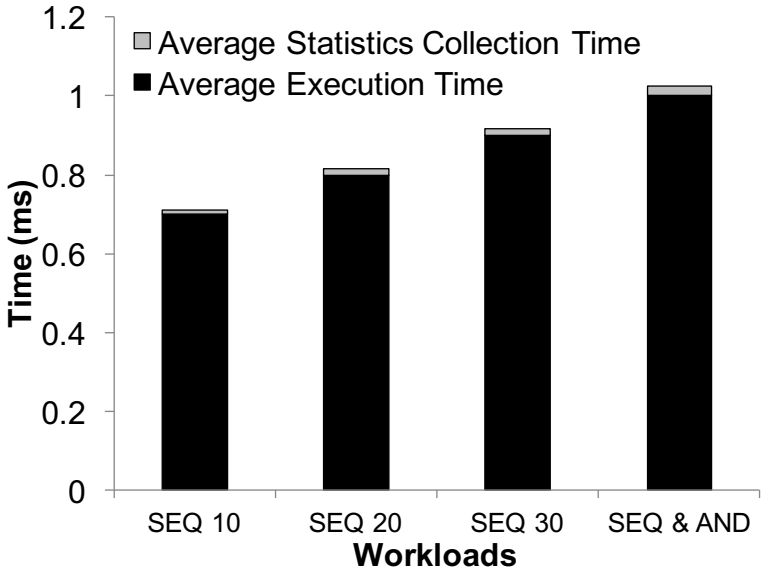


Vary the length of queries

Experimental Results – Runtime

- Average execution time
  - SPASS outperforms prefix-share, suffix-share and random sub-expression sharing approaches
  - SPASS statistics collection introduces negligible overheads





Conclusions

- Establish a model for sharing benefit estimation
  - Capturing overlapping event ordering among pattern queries
  - Capturing inter-query event correlation
- SPASS optimizer identifies shared sub-patterns in polynomial-time with proven bounds on optimality
- SPASS runtime facilitates shared processing of common sub-patterns with shared continuous sliding views

Contact: Chuan Lei (chuan@nec-labs.com)