DIPSUM: Distributed Pattern Summaries for Efficient CEP Aggregates

Steven Purtzel[1], Samira Akili[2], Marc Sebastian Kühne[1], and Matthias Weidlich[1]
Humboldt-Universität zu Berlin[1] & BIFOLD, TU Berlin[2]

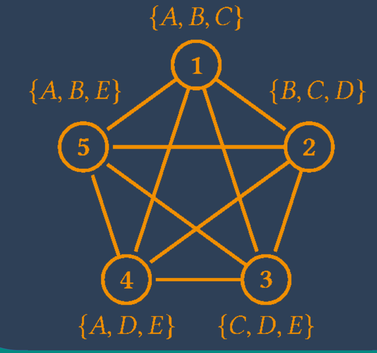
purtzesc@hu-berlin.de

Problem Setting

Evaluate CEP aggregate query...

SEQ(A,B,C+,D,E)
WITHIN 1 min
COUNT

...in event-network...



Rates:
r(A) > r(B),
r(C) ≫ OTHERS,
r(E) ≫ r(D)

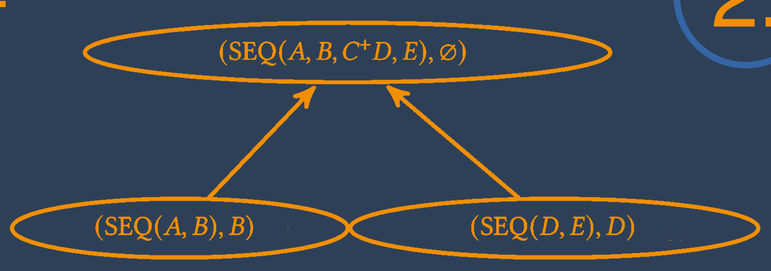
...such that:

- aggregate function is evaluated
- network transmission costs are minimized

DIPSUM Framework

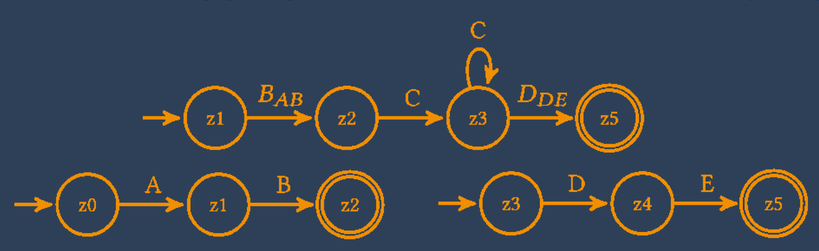
1. Query Decomposition:

- based on event type rates
- establish cut event types (event types connecting sub-queries)
- decompose query into sub-queries



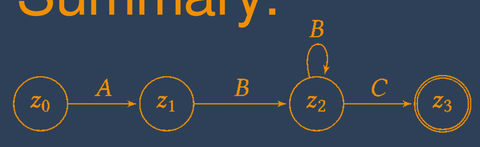
2. (Aggregator) NFA Derivation:

- derive (aggregator) NFA for each sub-query



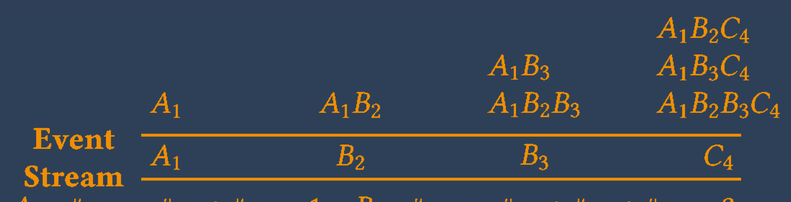
3. Partial State Summary:

- for each (aggregator) NFA derive a partial State Summary
- example: SEQ(A,B+,C)



$\#z_0 = 1; \#z_1 = \#z_2 = \#z_3 = 0$;
$A : \#z_1 \leftarrow \#z_1 + \#z_0$;
$B : \#z_2 \leftarrow \#z_2 + \#z_2 + \#z_1$;
$C : \#z_3 \leftarrow \#z_3 + \#z_2$

- count matches per state
  - ➜ global state counters
  - ➜ initialize: (1,0,0,0)
- count matches per event per state
  - ➜ local state counters
  - ➜ represent partial aggregates
  - ➜ e.g., #matches B occurred in SEQ(A,B)
- derive count rules; state counter updates



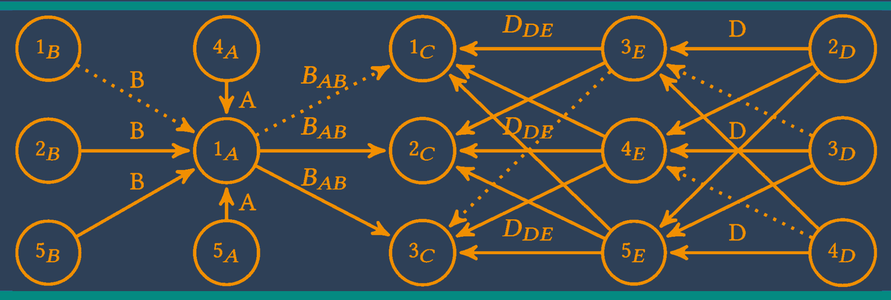
$A_1 : \#z_1 \leftarrow \#z_1 + \#z_0 = 1$ $B_3 : \#z_2 \leftarrow \#z_2 + \#z_2 + \#z_1 = 3$

$B_2 : \#z_2 \leftarrow \#z_2 + \#z_2 + \#z_1 = 1$ $C_4 : \#z_3 \leftarrow \#z_3 + \#z_2 = 3$

Global State Counters
(1,0,0,0) A_1➜ (1,1,0,0) B_2➜ (1,1,1,0) B_3➜ (1,1,3,0) C_4➜ (1,1,3,3)
Local State Counters B_2
(0,0,1,0) B_3➜ (0,0,2,0) C_4➜ (0,0,2,2)

4. Partial Summary Placement & Partial Aggregate Exchange:

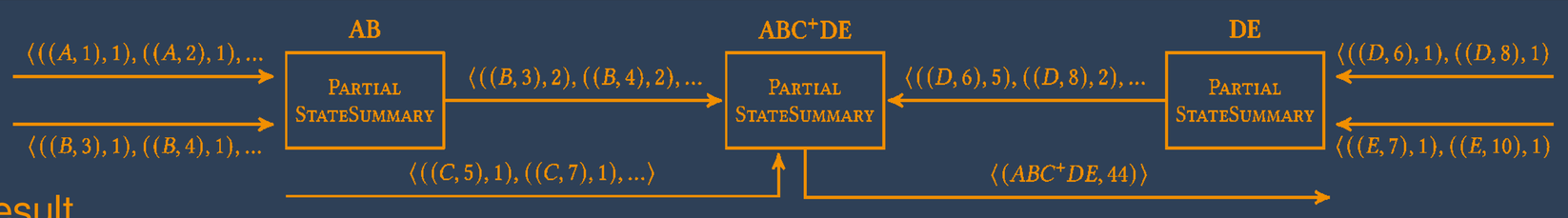
- based on rates & #nodes producing an event type



- leveraging inequalities for single- and multi-sink placements

5. Distributed State Summary:

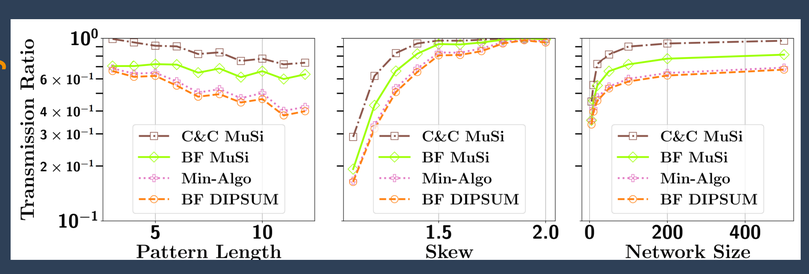
- stitching partial aggregates; obtain result



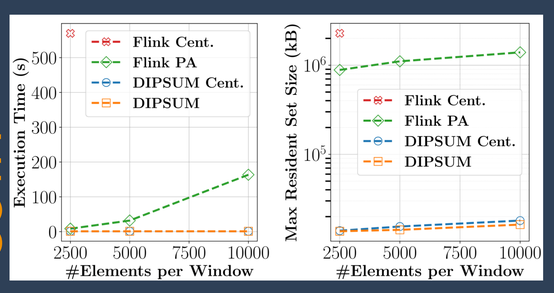
(Some) Evaluation Results

- synthetic data
- varying parameters
- transmission ratio
  - ➜ prod. cost/cent. cost
- FlinkCEP baselines
- simul. environment
- execution time & memory usage

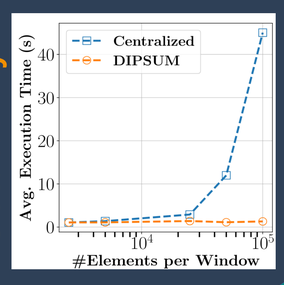
Sensitivity



SOTA



Scalability



Take Away

- DIPSUM plans **decompose** queries into sub-queries, **derive** and **place** pattern summaries, and **coordinate** partial aggregate exchange
- DIPSUM improves **transmission costs**, **execution times**, **throughput**, and **memory usage** by multiple orders of magnitude