

# Material in Response to Comments on: “DecoPa: Query Decomposition for Parallel Complex Event Processing”

November 30, 2023

## 1 Results for Network/Shuffle Costs

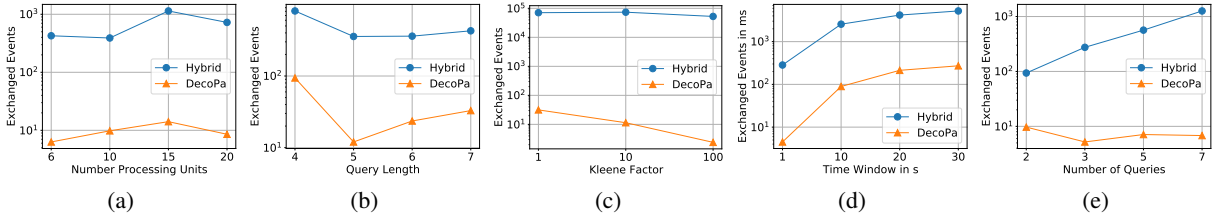


Figure 1: Exchanged events between processing units for DecoPa and Hybrid for maximal scaling factor of Hybrid approach.

## 2 Results for Optimization Time

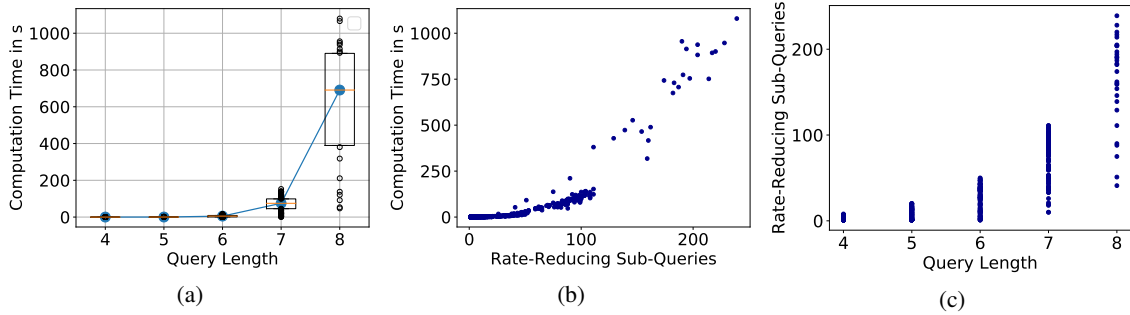


Figure 2: Computation time in seconds in (a) relation to query length and (b) number of sub-queries considered for decomposition. (c) Number of considered sub-queries per query length.

## 3 Query Snippets of Real-World Data Sets

**Citi Bike Types.** Characterization for derivation of event types:

- (1) short trip < 100s

- (2) long trip  $100s - 5000s$
- (3) very long trip  $> 5000s$
- (4) old, year of birth  $\leq 1962$
- (5) young, year of birth  $1962$
- (6) customer (instead of member) if not year of birth given

Examples for Event Types: LongC corresponds to a long trip with the driver being a customer. ShortY corresponds to a short trip with the driver being a young member. VLongO corresponds to a very long trip with the driver being a member of age group old.

**Google Cluster Types.** A description of the event types contained in the Google Cluster queries can be found at <https://drive.google.com/file/d/10r6cnJ5cJ89fPWCgj7j4LtLBqYN9RiI9>.

---

Query 1 - Citi Bike:

```
PATTERN AND (ShortY sy, VLongO vlo, ShortO so, LongO lo, LongC lc)
WHERE  $\forall (i, j) \in \{sy, vlo, so, lc\} \times \{sy, vlo, so, lc\}, i \neq j \wedge dist(i.startLoc, j.startLoc) \geq 6km$ 
WITHIN 24h
```

Query 2 - Citi Bike:

```
PATTERN AND (LongY ly, KL(ShortY sy), ShortO so, LongC lc)
WHERE  $\forall (i, j) \in \{ly, sy, so, lc\} \times \{ly, sy, so, lc\}, i \neq j \wedge dist(i.startLoc, j.startLoc) \geq 6km$ 
WITHIN 24h
```

Query 3 - Google Cluster:

```
PATTERN AND (Submit s, SEQ(Evict e1, Enable e2), Finish f)
WHERE  $\forall (i, j) \in \{s, e1, e2, f\} \times \{s, e1, e2, f\}, i.mem\_usage \geq j.mem\_usage \wedge i.cpu\_usage \geq j.cpu\_usage$ 
WITHIN 10min
```

Query 4 - Google Cluster:

```
PATTERN AND (KL(Schedule s), Queue l, Lost q)
WHERE  $\forall (i, j) \in \{s, l, q\} \times \{s, l, q\}, i.mem\_usage \geq j.mem\_usage \wedge i.cpu\_usage \geq j.cpu\_usage$ 
WITHIN 10min
```

---

Figure 3: Examples of the queries used in the evaluation for the Citi Bike and Google Cluster traces.