OVERLAP INTERVAL PARTITION JOIN

Anton Dignös, Michael H. Böhlen, and Johann Gamper
{adignoes, boehlen}@ifi.uzh.ch, gamper@inf.unibz.it

GOAL AND APPROACH

- Efficient interval partitioning for the overlap join in valid-time databases
- Find all pairs of tuples with overlapping intervals
- Partition intervals according to position and duration
- Self-adjusting: automatically determine the optimal number of partitions

OVERLAP INTERVAL PARTITIONING - OIP

- Time range is divided into k granules of equal duration
- Partitions are any sequence of contiguous granules

\[
k = 3
\]

Low k ⇒ fewer partition accesses (less overlapping boxes)

\[
k = 4
\]

High k ⇒ more precise partitions (better fitting boxes)

Constant clustering guarantee: Duration of tuple and partition differs by less than two granules.

OIPJOIN

Algorithm
1. Determine parameter k for OIP
2. Partition both input relations using k
3. Join tuples within overlapping partitions

Determining Parameter k for the OIPJoin

Approach: Find k by minimize the overhead cost function \( \text{cost}(k) \) w.r.t. k:

- partially filled blocks
  \( (1 \text{ trailing block per partition}) \)
- search in access structure
  \( (2 \text{ comparisons in access list}) \)

\[
\text{cost}(k) = |p_o| \cdot \text{APA} \cdot (c_i + 2 \cdot c_{cpu}) +
\]

- cost for partition accesses
- cost for false hits

more data is fetched
(1 false hit within a block)

identifying and discarding
(2 comparisons per false hit)

Result: \( k = f(\text{CPU cost, IO cost, relation sizes}) \)

EMPIRICAL EXPERIMENTS

Cost function compared to Runtime

Impact of long tuples

Personnel and file change data

Summary

- OIP intervals according to position and duration.
- \( OIP \) partitions are more efficient than \( OIP \).
- \( OIP \) is self-adjusting: k is determined by minimizing the total cost for false hits and partition accesses.
- \( OIP \) is robust for long tuples.

Future Work

- Advanced statistics to calculate the number of empty partitions and the reduced average number of partition accesses APA, for instance using histograms.
- Study the maintenance of \( OIP \).
- Refinement of cost function for different buffer replacement strategies.