Leveraging Range Joins for the Computation of Overlap Joins

Anton Dignös1, Michael H. Böhler2, Johann Gamper1, Christian S. Jensen3, and Peter Moser4

1Free University of Bozen-Bolzano, 2University of Zurich, 3Aalborg University, 4NITechpark Südiströ/Ald Age

OVERLAP JOIN
Given two relations containing periods, find pairs of tuples satisfying an equality predicate and overlap on periods [1].

SELECT ...
FROM emp CROSS JOIN dept d ON e.di = d.si AND e = OVERLAPS d.
P
Overlap join for equality on DNo and overlap on P

EName DNo [B E] DNo DName [B E]
Sum 2 1 6 1 HR 1 1
Ann 1 2 5 1 Test 1 6
Joe 2 4 8 2 QA 6 10
Sue 1 9 11 1 HR 2 11

EVALUATION USING RANGE JOINS

1. Transformation of the overlap predicate
Equivalence:
\[ r.P \text{ overlaps } s.P \iff r.B < s.E \land s.B < r.E \]
\[ \quad \equiv r.B \leq s.B < r.E \lor s.B < r.B < s.E \]

Properties:
- Two disjunctive range conditions (\( \equiv \) BETWEEN AND)
- Two conditions are disjoint (can be evaluated independently)

Rewrite:
SELECT ...
FROM emp CROSS JOIN dept d ON e.di = d.si AND e = s.d AND s.E = s.d;
UNION ALL
SELECT ...
FROM emp CROSS JOIN dept d ON e.di = d.si AND e = s.d AND s.E = s.d;

EXPERIMENTAL EVALUATION

Stand-alone (main memory) algorithm
- Without equality condition
- With equality condition
- On par with the state-of-the-art [3]
- More general algorithm

Index-based solution (in PostgreSQL)
- Without equality condition
- With equality condition
- Much faster compared to the state-of-the-art (e.g., [4])
- Clustering is very effective

CHALLENGES
- Overlap predicate consist of inequalities on 4 attributes
  \( \Rightarrow \) Specialized algorithms/indexes required
- Additional equality predicate needs to be supported
  \( \Rightarrow \) WHERE \( = \) AND \( \_ \_ \_ \_ \) overlaps \( \_ \_ \_ \_ \_ \)
- Different interval definitions should be supported
  \( \Rightarrow \) \([B,E],[B,E],[B,E],[B,E]\)

SUMMARY OF CONTRIBUTIONS
- We provide a new and simple rewriting of the overlaps predicate that transforms an overlap join into the union of two independent range joins.
- Our solution supports the combination of the overlaps predicate with non-temporal equality constraints.
- We provide a strict total order for period boundaries over discrete and continuous domains and prove its correctness. This enables support for all common interval definitions for period timestamps as well as relations where tuples might have period timestamps with different interval definitions.
- We show how to evaluate overlap joins in DBMSs by taking advantage of B-trees.
- We show how the rewriting can be used to devise an efficient yet simple main memory algorithm for overlap joins based on the sort-merge join paradigm.
- An extensive empirical evaluation shows that (a) our main memory algorithm performs on par with the state-of-the-art stand-alone competitors and that (b) the evaluation of the overlap join using B+ trees in an existing DBMS outperforms the state-of-the-art systems competitors.

REFERENCES

FUNDING
This work was funded by the Autonomous Province of Bozen-Bolzano Research “Sudtirol/Alto Adige 2019” through the project Enabling Industrial-Strength, Open-Source Temporal Query Processing – ISTeP and by the Innovation Fund Denmark centre, DIREC.

Published in: The VLDB Journal, Volume 31, Number 1, January 2022.
Presented at: 48th International Conference on Very Large Data Bases, Sydney, Australia (and hybrid), September 05–09, 2022.