Predicate Matching (1)

Problem:
- Data stream sharing requires identifying shareable data streams
- Identification process involves matching (selection/join) predicates
  Given a predicate \( p_1 \) and a (query) predicate \( p_2 \), does \( p_2 \) imply \( p_1 \)?
  If not, how can we alter \( p_1 \) for the implication to become valid?

Example Predicates:
  Stream Predicate \( p_1: \) \( (a \geq 3) \land (a \leq 12) \land (b \geq 0) \land (b \leq 5) \lor
  (a > 9) \land (a \leq 14) \land (b > 2) \land (b \leq 8) \lor
  (a < 0) \land (a \leq 5) \land (b > 1) \land (b \leq 6) \)
  Query Predicate \( p_2: \) \( (a \geq 1) \land (a \leq 8) \land (b > 2) \land (b \leq 4) \)

Quick Check (QC):
- Checks conjunctive subpredicates for containment
- Can be combined with any of the other matching algorithms

Heuristics with Simple Relaxation (HSR):
- Disjunctively adds conjunctive subpredicates of \( p_2 \) to \( p_1 \)
- Relaxes subpredicates of \( p_1 \) to contain subpredicates of \( p_2 \)
- Fast and easy to implement
- Misses matches in general causing unnecessary predicate relaxations
-Generally increases number of disjunctions in \( p_1 \)

Predicate Evaluation

Problem:
- Given a predicate \( p \) and a data item \( i \), does \( i \) satisfy \( p \)?
- Efficiently evaluate disjunctive predicates with potentially many disjunctions

Standard Evaluation (SE):
- Sequential scan
- Early exit when a match occurs

Index-based Evaluation (IE):
- Multi-dimensional index
- Early exit when a mismatch occurs

Optimization:
- Multi-dimensional index support (I) for predicate matching and evaluation
- Improve performance of evaluation index through short-circuiting (SC)

Benchmark Results

Predicate Matching:

- HCR+Q: Heuristics with Complex Relaxation (HCR):
  - Relaxes subpredicates of \( p_1 \) to contain subpredicates of \( p_2 \)
  - Relatively fast and easy to implement
  - Does not add any disjunctions to \( p_1 \)
  - Might add unnecessary parts of the data space to \( p_1 \)
  (leads to approximate results during predicate evaluation)

- EM: Exact Matching (EM):
  - Split algorithm
  - Exactly identifies matches, mismatches, and non-matching parts
  - Exponential complexity in number of subpredicates

- QC: Quick Check (QC):
  - Inapplicable for larger problem sizes
  - Use heuristics instead

Predicate Evaluation:

- SE: Standard Evaluation (SE):
  - Early exit when a match occurs

- IE: Index-based Evaluation (IE):
  - Early exit when a mismatch occurs

- HSR+Q: Heuristics with Simple Relaxation (HSR):
  - Inapplicable for larger problem sizes

StreamGlobe Basics:
- StreamGlobe: Distributed Data Stream Management System (DSMS)
- Super-Peers process and route data streams
- Thin-Peers publish and subscribe to data streams

Benefits of Data Stream Sharing in StreamGlobe:
- Stream sharing avoids redundant stream transmission
- Sharing computational results avoids redundant computation
- Early filtering and aggregation avoid unnecessary data transmission

- Reduced network traffic
- Reduced peer load