

## Abstract

Emerging data stream processing applications present new challenges that are not addressed by traditional DBMS technologies. To provide practical solutions for matching highly dynamic data streams with multiple long-lived and dynamically-updated continuous queries, a stream processing system should support incremental evaluation over new data, query optimization for continuous queries including computation sharing among multiple queries.

This thesis addresses these problems, presents the solutions in a prototype called ARGUS, and conducts experimental evaluations on the implemented techniques. Incremental query evaluation is realized by a set of algorithms based on materializing intermediate results to incrementally evaluate selections/joins (Rete), aggregates (incremental aggregation), and set operators (incremental set operations). The query optimization techniques include transitivity inference to derive highly selective predicates, conditional materialization to selectively materialize intermediate results, join order optimization to reduce join computations, and minimum column projection to project only necessary columns. Computation sharing is realized by an incremental multiple query optimization (IMQO) approach for tractable plan construction and dynamic query registration. It applies four steps to register a new query  $Q$ , recording existing query computations of the multi-query plan  $R$ , searching common computations between  $Q$  and  $R$ , selecting optimal sharing paths, and adding new computations to obtain final results for  $Q$  and  $R$ . The thesis presents a comprehensive computation indexing and searching scheme, and presents several sharing strategies. Finally, the evaluations on two data sets show that each technique leads to significant improvement in system performance up to hundreds-fold speed-up.

ARGUS is implemented atop a widely used commercial DBMS Oracle to allow fast deployment of the prototype as a value-added package to existing database applications where requirements of stream processing are growing rapidly in both scale and diversity. Future work includes supporting adaptive query processing, supporting distributive and parallel computing, and execution optimization.

Keywords: Stream Data, Continuous Query, Incremental Multiple Query Optimization, Rete, Incremental Query Evaluation, Transitivity Inference, Database, Conditional Materialization, Predicate Set, Extended Predicate Set Operation, Canonical Predicate Form, Predicate Indexing, Computation Sharing.