Scalable Numerical Queries by Algebraic Inequality Transformations

UDBL
Uppsala Database Laboratory

Thanh Truong, Tore Risch
{Thanh.Truong, Tore.Risch}.it.uu.se
Uppsala University, Sweden

Introduction

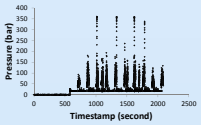
SCENARIO

- Data streams from machines are logged and bulk loaded into relational DBMS
- SQL queries to find suspected deviations from normal behavior based on measured sensor values (mv).
- They involve numberical inequality conditions.
- Ordered indexes on mv *idx(mv)* are added to improve the search.

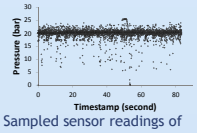
MOTIVATION

- Query optimzer cannot utilize the presence of ordered indexes.
- It requires manual query reformaltion because no automical solution exists

➔ Our solution: **Algebraic Query Inequality Transformation**



Sampled sensor readings of class A



Sampled sensor readings of class B

Contributions

- Algebraic query transformation strategy (AQIT) on numerical SQL queries,
  - AQIT is automatically and transparent to the user
  - AQIT substantially improves query performance
- The prototype system SLAS (Stream Log Analysis System) implements AQIT as a SQL pre-processor to a relational DBMS.

Example

Query Q identifies abnormal behaviors based on absolute deviations:

"When and for what machines did the pressure reading of measure class B deviate more than @thB from its expected value?"

Original SQL query before AQIT

```
SELECT vb.m, vb.bt, vb.et
FROM   measuresB vb, sensor s
WHERE  vb.m =s.m          AND
       vb.s=s.s           AND
   abs(vb.mv - s.ev) > @thB
```

Idx(measuresB.mv) is NOT exposed

Transformed SQL query after AQIT

```
SELECT vb.m, vb.bt, vb.et
FROM   measuresB vb, sensor s
WHERE  vb.m =s.m AND vb.s=s.s AND
     ((vb.mv > @thB +s.ev) OR
      (vb.mv < -@thB +s.ev))
```

Idx(measuresB.mv) is exposed

Stream Log Analysis System (SLAS)

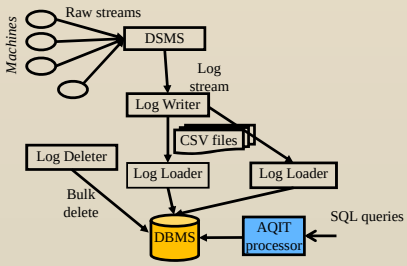


Fig1. Stream Log Analysis System

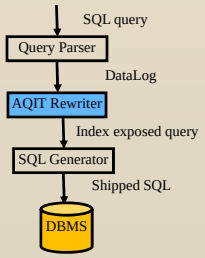


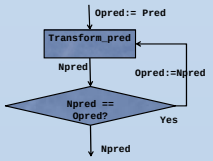
Fig2. AQIT Processor

- Process raw streams of measurements from different machines.
- *Log Writer* produces CSV file of tuples (m, s, bt, et, mv)
- *Log Loader* bulk loads the new log files into DBMS
- *Log Deleter* deletes log data from the DBMS according to user specified configuration parameters.

Algebraic Inequality Transformation

- Given
  - *mv* is a variable bound to indexed attribute A
  - *T* is transformable functions, currently T ∈ {+, -,/,*, power, sqrt, abs}
  - *F, F'* are functions composable from any combination of T
  - *ψ, ψ'* is an inequality comparision ψ, ψ' ∈ {>, <, ≤, ≥}
- Transform (F(mv) ψ ε) ⇒ (mv' ψ 'F'(ε)).

  Therefore, mv is exposed, which implies index on attribute A is exposed.
- The query retains if AQIT fails



- Fixpoint algorithm
- *Transform_pred* to expose indexes
- Until, no further indexes can be exposed

TRANSFORM_PRED

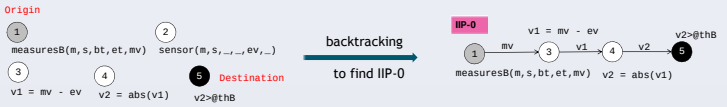
- *Chain()* finds some path between an indexed variable and an inequality predicate that can be exposed.
- *Expose()* transforms the found path so that the index becomes exposed
- *Substitute()* replaces predicates in the original pred with predicates in the new paths

```
function transform_pred (pred)
Input: A predicate pred
Output: A transformed predicate or the original pred
..
set path = chain(pred)
if path not null then
   set exposedpath = expose(path)
   if exposedpath not null then
      return substitute(pred, path, exposedpath)
   end if
end if
...
```

AQIT Pseudo Code
http://www.it.uu.se/research/group/udbl/aqit/AQIT_PseudoCode.pdf

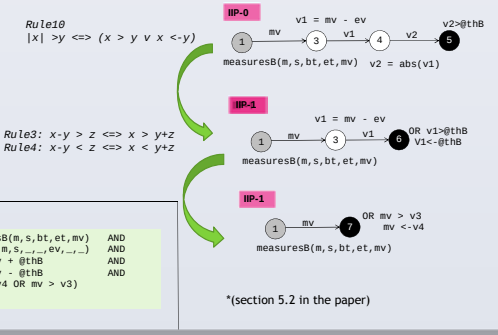
Step 1 - CHAIN

It produces a path of predicates that links on indexed variable with one inequality predicate. Such path is called as indexed inequality path IIP



Step 2- EXPOSE

Applying algebraic rules(*) to shorten the IIP until there is no intermediate nodes



Step 3 - SUBSTITUTE

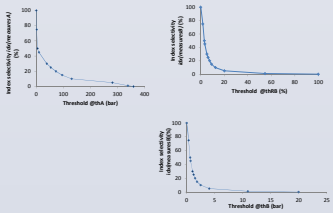
BEFORE		AFTER	
measuresB(m,s,bt,et,mv)	AND	measuresB(m,s,bt,et,mv)	AND
sensor(m,s,_,_,ev,_)	AND	sensor(m,s,_,_,ev,_)	AND
v1 = mv – ev	AND	v3 = ev + @thB	AND
v2 = abs(v1)	AND	v4 = ev - @thB	AND
v2 > @thB		(mv < v4 OR mv > v3)	

*(section 5.2 in the paper)

Experiment

- SQL Server™ 2008 R2 as backend DBMS
- Simulated data from industrial log files
- Studied mapping between thresholds and selectivities
- Experiments were conducted with and without AQIT preprocessing, with different settings A, B, and C

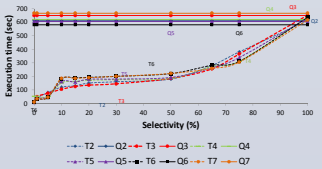
Mapping between thresholds and selectivities



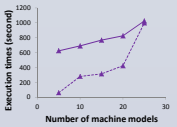
A-Varying database sizes

- **6 benchmarck queries**
- **Different settings**
- http://www.it.uu.se/research/group/udbl/aqit/Benchmark_Queries.pdf

B-Varying selectivies



C-Varying machines



EVALUATION

- AQIT improved the performance of the benchmark queries substantially.
- AQIT exposes hidden indexes while the backend DBMS decides whether to utilize them or not
- AQIT does not make queries slower

**The work was supported by the Smart Vortex EU project