Tiresias
The Database Oracle for How-To Queries

Alexandra Meliou
Dan Suciu

§University of Massachusetts Amherst
*University of Washington
Hypothetical (What-if) Queries

Example from [Balmin et al. VLDB’00]:
“An analyst of a brokerage company wants to know what would be the effect on the return of customers’ portfolios if during the last 3 years they had suggested Intel stocks instead of Motorola.”

change something in the source (hypothesis)  forward  observe the effect in the target
Modified example:
“An analyst wants to ask **how to** achieve a 10% return in customer portfolios, with the least number of trades.”

- find changes to the source that achieve the desired effect
- **reverse**
- declare a desired effect in the target
TPC-H example

- A manufacturing company keeps records of inventory orders in a LineItem table.
  - KPI: Cannot order more than 7% of the inventory from any single country

(variables)
- Can reassign orders to new suppliers as long as the supplier can supply the part

(constraints)
- Minimize the number of changes

(optimization objective)
Constraint Optimization on Big Data

this is for a set of 10 lineitems and 40 suppliers

MathProg

Impractical!

Mixed Integer Programming (MIP) solver

transform into data updates

DB
Demo: Tiresias

a tool that makes how-to queries practical
**Tiresias: How-To Query Engine**

**TiQL (Tiresias Query Language)**

**RULES:**

- `HChooseS(ok,pk,sk,qnt,sk',country)`  
  `:- PartSupp(pk,sk') & LineItem(ok,pk,sk,qnt)`  
  `& SuppNation(sk2,country)`

- `HLineItem(ok,pk,sk',qnt)  `  
  `:- HChooseS-ok,pk,sk,qnt,sk',country)`

- `HOrderSum(country,count(*))  `  
  `:- HChooseS-ok,pk,sk,qnt,sk',country)`

- `[c? <= 10] <- HOrderSum(country,c?)`

- `MAXIMIZE(count(*))  `  
  `:- HChooseS-ok,pk,sk,qnt,sk,country)`
Overview

Visualizations

TiQL

MathProg or AMPL
**Overview**

**Visualizations**

**TiQL**

- Language semantics
- Evaluation of a TiQL program: Translation from TiQL to linear constraints
- Performance optimizations

**MathProg or AMPL**

**Demo**
Tiresias Query Language

- Datalog-like notation:
  \[ P(\bar{x}) \colon \neg B_1(\bar{x}_1) \land B_2(\bar{x}_2) \land \cdots \land B_n(\bar{x}_n) \]

  - head
  - body: conjunction of predicates

- TiQL semantics:
  Mapping from EDBs (Extensional Database) to possible worlds over HDBs (Hypothetical Database)

  \[ \text{HDB} \rightarrow HP(\bar{x}) \colon \neg \text{body} \]
## TiQL Rules

### Deduction Rule

**Symbol:** \[ HP(\bar{x}) \ :- \ \text{body} \]

**Semantics:**
Similar to repair-key semantics [Antonova et al. SIGMOD’07], [Koch ICDT’09]

### Reduction Rule

**Symbol:** \[ HP(\bar{x}) \ :- \ \text{body} \]

**Semantics:**
Takes a subset of tuples

### Constraint Rule

**Symbol:** \[ \text{[arithm-pred]} \ :- \ \text{body} \]

**Semantics:**
The head predicate needs to hold for all tuples
Overview

Visualizations

Demo

RULES:
- HChooseS(ok,pk,sk,qnt,sk',country) :- PartSupp(pk,sk') & LineItem(ok,pk,sk,qnt)
  & SuppNation(sk2,country)
- HLineItem(ok,pk,sk',qnt) :- HChooseS(ok,pk,sk,qnt,sk',country)
- HOrderSum(country,count(*)) :- HChooseS(ok,pk,sk,qnt,sk,country)
[c? <= 10] <- HOrderSum(country,c?)

MAXIMIZE(count(*)) :- HChooseS(ok,pk,sk,qnt,sk,country)

Language semantics

Evaluation of a TiQL program: Translation from TiQL to linear constraints

Performance optimizations

TiQL

MathProg or AMPL
Evaluating a TiQL Program

TiQL

RULES:

\[
\text{H\text{ChooseS}}(\text{ok}, \text{pk}, \text{sk}, \text{qnt}, \text{sk'}, \text{country}) \leftarrow \text{PartSupp}(\text{pk}, \text{sk'}) \& \text{LineItem}(\text{ok}, \text{pk}, \text{sk}, \text{qnt}) \& \text{SuppNation}(\text{sk2}, \text{country})
\]

\[
\text{H\text{LineItem}}(\text{ok}, \text{pk}, \text{sk'}, \text{qnt}) \leftarrow \text{H\text{ChooseS}}(\text{ok}, \text{pk}, \text{sk}, \text{qnt}, \text{sk'}, \text{country})
\]

\[
\text{H\text{OrderSum}}(\text{country}, \text{count(*))} \leftarrow \text{H\text{ChooseS}}(\text{ok}, \text{pk}, \text{sk}, \text{qnt}, \text{sk'}, \text{country}) \& \text{[c? <= 10]} \leftarrow \text{H\text{OrderSum}}(\text{country}, \text{c?})
\]

\[
\text{MAXIMIZE}((\text{count(*))}) \leftarrow \text{H\text{ChooseS}}(\text{ok}, \text{pk}, \text{sk}, \text{qnt}, \text{sk}, \text{country})
\]
Evaluating a TiQL Program

**HChooseS**(ok, pk, sk, sk') :- PartSupp(pk, sk') & LineItem(ok, pk, sk, qnt)
**Key Constraints**

**CORE_HChooseS**

<table>
<thead>
<tr>
<th>ok</th>
<th>pk</th>
<th>sk</th>
<th>sk'</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>P15</td>
<td>S10</td>
<td>S10</td>
</tr>
<tr>
<td>1</td>
<td>P15</td>
<td>S10</td>
<td>S21</td>
</tr>
<tr>
<td>2</td>
<td>P32</td>
<td>S43</td>
<td>S10</td>
</tr>
<tr>
<td>2</td>
<td>P32</td>
<td>S43</td>
<td>S43</td>
</tr>
</tbody>
</table>

\[ x_1 \]
\[ x_2 \]
\[ x_3 \]
\[ x_4 \]

**Key(ok, pk, sk)**

- \[ x_1 + x_2 \leq 1 \]
- \[ x_3 + x_4 \leq 1 \]

**NOT a possible world**

<table>
<thead>
<tr>
<th>ok</th>
<th>pk</th>
<th>sk</th>
<th>sk'</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>P15</td>
<td>S10</td>
<td>S10</td>
</tr>
<tr>
<td>1</td>
<td>P15</td>
<td>S10</td>
<td>S21</td>
</tr>
</tbody>
</table>

\[ \forall k_j : \sum_{i, key(x_i) = k_j} x_i \leq 1 \]

\[ 0 \leq x_i \leq 1 \]
Provenance Constraints

- A TiQL rule specifies transformations
- Transformations define provenance
  - Boolean semantics for queries without aggregates
  - Semi-module provenance for queries with aggregates [Amsterdamer et al. PODS’11]

\[
\text{Disjunction: } Y = X_1 \lor X_2 \lor \ldots \lor X_n \\
\forall i, y \geq x_i \\
y \leq \sum_i x_i
\]

\[
\text{Conjunction: } Y = X_1 \land X_2 \land \ldots \land X_n \\
\forall i, y \leq x_i \\
y \geq \sum_i x_i - (n - 1)
\]
Demo

MathProg or AMPL

Exercise

AMPL

TiQL

HChooseS(ok, pk, sk, qnt, sk', country) :- PartSupp(pk, sk') & LineItem(ok, pk, sk, qnt) & SuppNation(sk2, country)

HLineItem(ok, pk, sk', qnt) :- HChooseS(ok, pk, sk, qnt, sk', country)

HOrderSum(country, count(*)) :- HChooseS(ok, pk, sk, qnt, sk', country)

[c? <= 10] <- HOrderSum(country, c?)

MAXIMIZE(count(*)) :- HChooseS(ok, pk, sk, qnt, sk, country)

Overview

Visualizations

Demo

Language semantics

Evaluation of a TiQL program: Translation from TiQL to linear constraints

Performance optimizations
Optimizing Performance

- **Model optimizer**
  - eliminates variables, constraints, and parameters
  - uses *key constraints, functional dependencies, and provenance*

  Significantly faster than letting the MIP solver do it

- **Partitioning optimizer**

  \[
  \begin{align*}
  \text{max} & \quad (x_1 + x_2 + x_3 + x_4) \\
  \text{s.t.} & \quad x_1 + x_2 \leq 50 \\
  & \quad x_3 + x_4 \leq 50 \\
  & \quad x_i \geq 0 \\
  \end{align*}
  \]

  \[
  \begin{align*}
  \text{max} & \quad (x_1 + x_2) \\
  \text{s.t.} & \quad x_1 + x_2 \leq 50 \\
  & \quad x_i \geq 0 \\
  \end{align*}
  \]

  \[
  \begin{align*}
  \text{max} & \quad (x_3 + x_4) \\
  \text{s.t.} & \quad x_3 + x_4 \leq 50 \\
  & \quad x_i \geq 0 \\
  \end{align*}
  \]
Evaluation of the Model Optimizer

Runtime (sec) vs. Data size (# of tuples)
Evaluation of Tiresias Partitioning

complex dependency on the granularity of partitioning

granularity of partitioning
The MIP solver runtime (per partition) does not increase with data size.

Constructor time depends on DB query execution time.
Related Work

- **Provenance**
  - [Amsterdamer et al. PODS’11], [Cui et al. TODS’00], [Green et al. PODS’07]

- **Incomplete databases**
  - [Antonova et al. SIGMOD’07], [Imielinski et al. JACM’84], [Koch ICDT’09]

- **Other RDM problems**
  - [Arasu et al. SIGMOD’11], [Binnig et al. ICDE’07], [Bohannon et al. PODS’06], [Fagin et al. JACM’10]
Next Steps with Tiresias

- Handling non-partitionable problems
- Approximations
- Parallelization and handling of skew
- Result analysis and feedback-based problem generation
SIGMOD Demo Group C

Location: Vaquero A
Time: 13:30-15:00
Contributions

- How-To queries
- Using MIP solvers to answer How-To queries
- Tiresias prototype implementation

http://db.cs.washington.edu/tiresias