TPC-BiH: A Benchmark for Bitemporal Databases

Martin Kaufmann, Peter M. Fischer, Norman May, Donald Kossmann
### Motivation: Inventory Management

**OrderedItems**

<table>
<thead>
<tr>
<th>OrderID</th>
<th>ProdID</th>
<th>Ship</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>mbpro</td>
<td>false</td>
<td>10 – ∞</td>
</tr>
<tr>
<td>1</td>
<td>w530</td>
<td>false</td>
<td>16 – 19</td>
</tr>
<tr>
<td>2</td>
<td>mbair</td>
<td>false</td>
<td>17 – ∞</td>
</tr>
<tr>
<td>1</td>
<td>w530</td>
<td>true</td>
<td>19 – ∞</td>
</tr>
</tbody>
</table>

**Inventory**

<table>
<thead>
<tr>
<th>ProdID</th>
<th>Price</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>w530</td>
<td>3000</td>
<td>1 – 15</td>
</tr>
<tr>
<td>mbpro</td>
<td>2000</td>
<td>1 – 15</td>
</tr>
<tr>
<td>mbair</td>
<td>2500</td>
<td>1 – ∞</td>
</tr>
<tr>
<td>w530</td>
<td>3050</td>
<td>15 – ∞</td>
</tr>
<tr>
<td>mbpro</td>
<td>2100</td>
<td>15 – 17</td>
</tr>
<tr>
<td>mbpro</td>
<td>1900</td>
<td>17 – ∞</td>
</tr>
<tr>
<td>x220</td>
<td>2300</td>
<td>20 – ∞</td>
</tr>
</tbody>
</table>

---

### Analysing temporal data

1) What was the total value of the inventory at each point in time?
   - **Temporal Aggregation**

2) What was the price of MacBook Pro on January 15th?
   - **Time-Travel**

3) What was the most expensive order last year?
   - **Temporal Join**
Motivation: Why do we need another benchmark?

- Increasing demand for temporal features
  - Dozens of apps require it
  - Adoption of bitemporal features in SQL:2011
  - Support by commercial database vendors

- Users need to assess performance
  - Keeping and querying history not reflected by existing benchmarks
  - Temporal processing is performance-critical
  - Understanding of the performance characteristics of alternative implementations of temporal operators
Outline

- Motivation
- State-of-the-art
- The TPC-BiH Benchmark
- Preliminary Results
- Conclusion
Temporal Data: State of the Art

- Long-Running Research

- Currently limited support on language side
  - SQL:2011 is a big step forwards
  - No complete coverage of all use cases

- Commercial systems start adding temporal features
  - Bitemporal data: Oracle, Teradata, DB2 (SQL:2011)
  - System time: SAP HANA
Existing Temporal Benchmarks

- Previous work on benchmarking the temporal dimension:
  - Requirements specifications (Dunham et al., 1995)
  - Functional tests (test suite from the TSQL2 editors)

- Recent activity and closest match:
  "Adding a Temporal Dimension to the TPC-H Benchmark" (TPCTC 2012)
  - Limited extension of TPC-H schema
  - Sketches of data and queries
  - Focus on TSQL2
Outline

- Motivation
- State-of-the-art
- The TPC-BiH Benchmark
- Preliminary Results
- Conclusion
Methodology of the Benchmark

Overall goals

- Comprehensive benchmark for bitemporal query processing
- Benchmark settings reflect
  - real-life customer workloads
  - synthetic tests of temporal properties
- Benchmark is targeted towards SQL:2011

Schema

- Extend TPC-H tables with different types of history classes:
  - fully bitemporal
  - degenerated
  - multiple user times
Methodology of the Benchmark (II)

Data

- Initialization
  - Initial version is regular TPC-H data
  - Derive application time from TPC-H data semantics

- History
  - Evolve through carefully designed update scenarios
  - Maintain overall distributions at each point in time

Queries

- Coverage of common temporal DB requirements
- Stressing the system for individual time dimensions
- Correlations among the dimensions whenever relevant
Data Generator

- Use TPC-H dbgen for version 0
- Generate history by adding 9 realistic update scenarios:

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Updated Tables</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Order</td>
<td>Customer, Orders, Lineitem</td>
<td>0.3</td>
</tr>
<tr>
<td>Cancel Order</td>
<td>Orders, Lineitem</td>
<td>0.05</td>
</tr>
<tr>
<td>Deliver Order</td>
<td>Customer, Orders, Lineitem, Partsupp</td>
<td>0.25</td>
</tr>
<tr>
<td>Receive Payment</td>
<td>Orders, Customer</td>
<td>0.20</td>
</tr>
<tr>
<td>Update Stock</td>
<td>Partsupp</td>
<td>0.05</td>
</tr>
<tr>
<td>Delay Availability</td>
<td>Part</td>
<td>0.05</td>
</tr>
<tr>
<td>Change Supplier Price</td>
<td>Partsupp</td>
<td>0.05</td>
</tr>
<tr>
<td>Update Supplier</td>
<td>Supplier</td>
<td>0.049</td>
</tr>
<tr>
<td>Manipulate Order</td>
<td>Orders, Lineitems</td>
<td>0.01</td>
</tr>
</tbody>
</table>
Queries

- The queries cover operations such as time travel, key in time, temporal joins and temporal aggregation

- Investigate many patterns of storage access and time- vs. key-oriented access with varying ranges and selectivity

- **Classes of TPC-BiH benchmark queries:**
  1) Pure-Timeslice Queries (Time Travel)
  2) Pure-Key Queries (Audit)
  3) Range-Timeslice Queries
  4) Bitemporal Queries
1) Pure-Timeslice Queries (Time Travel)

Motivation

- Testing "slices" of time, i.e., state of table at a previous time
- Each time dimension can be treated as a point or slice
  - Point: looking at a single time
  - Complete slice: full evolution of a tuple through a time dimension
- Evaluate on application time, system time or both

Example: "Value of all unshipped orders at a given time"

```
SELECT SUM(o_totalprice) AS revenue
FROM ORDERS
FOR SYSTEM_TIME AS OF TIMESTAMP '[TIME1]'
FOR BUSINESS_TIME AS OF '[TIME2]'
WHERE o_orderstatus = 'O'
```
2) Pure-Key Queries (Audit)

Motivation

- History of a single or small set of tuples
- Investigate how tuples evolve over time
- Evolution along the system time, application time(s) or both

Example: "What is the history of a given order?"

```sql
SELECT orderstatus, totalprice, orderpriority, shippriority, sys_time_start
FROM orders
FOR SYSTEM_TIME FROM '0001-01-01' TO '9999-12-30'
WHERE o_orderkey = [ORDER_KEY]
ORDER BY sys_time_start
```
3) Range-Timeslice Queries

Motivation

- Permit any combination of constraints
- General access pattern: both value and temporal aspects

Example: "When did we have the largest number of unshipped items?"

```sql
CREATE VIEW unshipped AS
    SELECT COUNT(*) AS total, li.VERSION() as version
    FROM LINEITEM li
    WHERE li.L_LINESTATUS = 'O'
    GROUP BY li.VERSION();

SELECT version, total FROM unshipped
WHERE total = (SELECT MAX(total) FROM unshipped)
```
4) Bitemporal Queries

Motivation

- Stress several time dimensions at the same time
- Vary the usage of each time dimension
  a) point in time: current vs. past
  b) sequenced/time range
  c) non-sequenced/agnostic of time
- Complementary query variants to cover all combinations
Outline

- Motivation
- State-of-the-art
- The TPC-BiH Benchmark
- Preliminary Results
- Conclusion
Evaluation Setup

- Out-of-the-box settings (tuning is ongoing work)
- Two DBMS tested
  - System A: RDBMS supporting bitemporal workloads
  - System B: Main memory DBMS supporting system time
- Run on single system
  - 2 Intel Xeon X5675 processors with 6 cores at 3.06 GHz
  - 192GB of DDR3-1066MHz RAM
- Data
  - Initial data scaling factor (as in TPC-H): h=1.0
  - History size m=1.0 (1 million update scenarios)
  - Update scenario: Transaction (no bulk loading, but generate data!)
- At this stage purely performance results, no deep analysis

Note: logarithmic y-axis for all plots!
Pure Timeslice Queries

Log(execution time [ms])

Server

System A

System B

T.1c (app)  T.1a (sys)  T.2c (app)  T.2a (sys)  T.3 (app)  T.3 (sys)  T.4 (app)  T.4 (sys)  T.5 Allversions  T.6 (app)  T.6 (sys)  T.6 Simulated App Time  T.7  T.8  T.9

h=1.0, m=1.0
Normal vs. Bitemporal

Log(execution time [ms])

h=1.0, m=1.0

System A Time Travel

System A Non-Temporal

Server

H.17  H.18  H.19  H.20  H.21  H.22
Simple Aggregation over Time

![Graph showing simple aggregation over time with system A and system B, and h=1.0, m=1.0.](image-url)
Range-Timeslice Queries

h=0.1, m=0.1
Outline

- Motivation
- State-of-the-art
- The TPC-BiH Benchmark
- Preliminary Results
- Conclusion
Ongoing and Future Work

- Better coverage of DBMSs
- (Basic) tuning for temporal workloads, deeper analysis of query plans
- Variance in workloads
- Impact of scaling and value distribution
- Evaluation of update performance
TPC-BiH: Conclusion

Properties
- Builds on existing benchmarks
- Comprehensive coverage of use cases for temporal data
- Data generator based on real-world scenarios

Preliminary Results for Commercial Systems
- Significant potential for optimization
- Not all common application use cases supported sufficiently
Covering Bitemporal Dimensions

![Bar chart showing execution time for different systems and servers.](chart.png)