Pathfinder: XQuery Compilation Techniques for Relational Database Targets

Jens Teubner · Technische Universität München

Joint work with: Torsten Grust, Peter Boncz, Martin Kersten, Maurice van Keulen, Stefan Manegold, Sjoerd Mullender, Jan Rittinger, Marc H. Scholl, …
Challenge: Construction of a Scalable XQuery Processor

XQuery:
- tree-structured XML data
- ordered sequences of items: \((x_1, \ldots, x_n)\)
- explicit iteration: for \(v \text{ in } e_1 \text{ return } e_2\)
- side effects: element \(t \{ e \}\)
Challenge: Construction of a Scalable XQuery Processor

XQuery:
- tree-structured XML data
- ordered sequences of items: \((x_1, \ldots, x_n)\)
- explicit iteration: \(\text{for } \$v \text{ in } e_1 \text{ return } e_2\)
- side effects: \(\text{element } t \{ e \}\)

Re-use existing RDBMS technology?
- flat, unordered data model: tables of tuples
- bulk-oriented processing
- no side effects
Challenge: Construction of a Scalable XQuery Processor

XQuery:
- tree-structured XML data
- ordered sequences of items: \((x_1, \ldots, x_n)\)
- explicit iteration: for \($v\ in e_1\ return\ e_2\)
- side effects: \(element\ t\ \{\ e\ \}\)

Re-use existing RDBMS technology?
- flat, unordered data model: tables of tuples
- bulk-oriented processing
- no side effects
This talk bridges the apparent gap.

trees, sequences, iteration, side effects

compositional compilation: loop-lifting

relational step evaluation: staircase join

tree encoding: XPath accelerator

tables of tuples, relational algebra, SQL

Pathfinder is a full open-source implementation of these techniques.
Pathfinder’s XML Storage is based on XPath Accelerator (Grust ’02)

Any encoding providing node identity/document order suffices.

We actually use a variant of this encoding: pre/size/level.
Relational XML storage can beat native XPath processors.

Use B-trees with \textit{low-selectivity} prefixes (e.g., \textit{level}, tag names)!
XPath is the backbone of every XQuery processor.

XPath:
- Context is a set of nodes
- Document order, duplicate-free result

Problems:
- Repeated scans over the same area
- Expensive sorting and duplicate elimination
XPath is the backbone of every XQuery processor.

XPath:
- Context is a set of nodes
- Document order, duplicate-free result

Problems:
- Repeated scans over the same area
- Expensive sorting and duplicate elimination

Staircase join: [VLDB 2003]
- Encapsulates tree awareness in a single join operator
- Cache-friendly and XPath-compliant
We injected staircase join into PostgreSQL 7.3.

Query: /descendant::age/ancestor::person

[Vldb 2004 Demo]

XML document size [MB]

Execution times

Page misses
XPath is only part the story.

- Variables and iteration: for $v$ in $e_1$ return $e_2$
- Sequence construction: $(e_1, e_2)$
- Element construction: `element { e_1 } { e_2 }`
- Dynamic typing: $e_1$ instance of $e_2$
- etc.

XQuery is a functional language, though.

- Process independent FLWOR iterations in parallel
- Use **bulk-oriented** processing capabilities of modern RDBMSs
Loop-lifting: Encode independent iterations using a single relation.

for $x$ in (1, 2, 3) return $x$ to 3

- Column $iter$ labels independent iterations.
- Sequence order is maintained in column $pos$.
- This is the **loop-lifted** encoding of an XQuery item sequence.
- The compilation procedure operates on loop-lifted sequence representations only.

<table>
<thead>
<tr>
<th>iter</th>
<th>pos</th>
<th>item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>
Loop-lifting: Encode independent iterations using a single relation.

for $x$ in (1, 2, 3) return $x$ to 3

- Column $iter$ labels independent iterations.
- Sequence order is maintained in column $pos$.
- This is the loop-lifted encoding of an XQuery item sequence.
- The compilation procedure operates on loop-lifted sequence representations only.

<table>
<thead>
<tr>
<th>iter</th>
<th>pos</th>
<th>item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>
Loop-lifting: Encode independent iterations using a **single** relation.

```
for $x$ in (1, 2, 3) return $x$ to 3
```

- Column *iter* labels independent iterations.
- Sequence order is maintained in column *pos*.
- This is the **loop-lifted** encoding of an XQuery item sequence.
- The compilation procedure operates on loop-lifted sequence representations only.

This representation is highly versatile.

- Item types to support **dynamic type** semantics
Loop-lifting: Encode independent iterations using a single relation.

```
for $x$ in (1, 2, 3) return $x$ to 3
```

- Column $iter$ labels independent iterations.
- Sequence order is maintained in column $pos$.
- This is the loop-lifted encoding of an XQuery item sequence.
- The compilation procedure operates on loop-lifted sequence representations only.

This representation is highly versatile.

- Item types to support dynamic type semantics
- Additional information to support, e.g., XQuery full-text search
Commodity RDBMSs readily provide all the functionality we need.

E.g., SQL on IBM DB2 Universal Database V 8.2.

![Execution time chart](image)
Pathfinder is a full implementation of a loop-lifting compiler.

- Fully compositional, in line with the XQuery language
Pathfinder is a full implementation of a loop-lifting compiler.

- **Fully compositional**, in line with the XQuery language

The resulting plans can be of significant size, though.
Pathfinder is a full implementation of a loop-lifting compiler.

- **Fully compositional**, in line with the XQuery language

The resulting plans can be of significant size, though.

To optimize relational plans, Pathfinder thus implements

1. **constant propagation**,
Pathfinder is a full implementation of a loop-lifting compiler.

- **Fully compositional**, in line with the XQuery language

The resulting plans can be of significant size, though.

To optimize relational plans, Pathfinder thus implements

1. constant propagation,
2. projection pushdown,
Pathfinder is a full implementation of a loop-lifting compiler.

- **Fully compositional**, in line with the XQuery language

The resulting plans can be of significant size, though.

To optimize relational plans, Pathfinder thus implements

1. **constant propagation**,  
2. **projection pushdown**,  
3. **functional dependency** and **data flow analyses**, and
Pathfinder is a full implementation of a loop-lifting compiler.

- Fully compositional, in line with the XQuery language

The resulting plans can be of significant size, though.

To optimize relational plans, Pathfinder thus implements

1. constant propagation,
2. projection pushdown,
3. functional dependency and data flow analyses, and
4. algebraic join detection.

(You saw these optimizations in yesterday’s demo session.)
Pathfinder targets the main-memory RDMBS MonetDB.

- Queries over multi-gigabyte XML instances answered in **interactive time** (XMark: 18 of 20 queries in \( \ll 1 \text{ min on } 1.1 \text{ GB} \))
- Unprecedented **scalability**
A complete and purely relational XQuery processing stack:

A relational tree encoding, derived from XPath accelerator, maps XML document trees into relational tables.

- Re-use of mature storage and indexing techniques
A complete and purely relational XQuery processing stack:

A relational tree encoding, derived from XPath accelerator, maps XML document trees into relational tables.

- Re-use of mature storage and indexing techniques

**Staircase join** encapsulates knowledge about our tree encoding in terms of a single join operator.

- Outstanding XPath performance on any RDBMS
A complete and purely relational XQuery processing stack:

A relational tree encoding, derived from XPath accelerator, maps XML document trees into relational tables.
- Re-use of mature storage and indexing techniques

Staircase join encapsulates knowledge about our tree encoding in terms of a single join operator.
- Outstanding XPath performance on any RDBMS

The loop-lifting compilation procedure maps arbitrary XQuery expressions to primitives of relational algebra.
- Implementation of iterative XQuery semantics in terms of efficient, bulk-oriented processing
Pathfinder is an ongoing, joint research project with CWI Amsterdam, U Twente, and U Konstanz.

- Algebraic optimization, cost and result size estimation
- New functionality: recursion, dynamic typing, and validation
- Alternative back-ends: Idefix (UKN), SQL:1999 [SIGMOD 2007]
Ongoing and Future Work

Pathfinder is an ongoing, joint research project with CWI Amsterdam, U Twente, and U Konstanz.

- Algebraic optimization, cost and result size estimation
- New functionality: recursion, dynamic typing, and validation
- Alternative back-ends: Idefix (UKN), SQL:1999 [SIGMOD 2007]

MonetDB/XQuery has started to spread across the world already.

- X-RPC: XQuery processing in peer-to-peer networks
- XIRAF: multi-hierarchical XML documents
- Tijah: full-text retrieval for the MonetDB/XQuery system
- ~150 SourceForge downloads per month (MonetDB/XQuery only)
Pathfinder is an ongoing, joint research project with CWI Amsterdam, U Twente, and U Konstanz.

- Algebraic optimization, cost and result size estimation
- New functionality: recursion, dynamic typing, and validation
- Alternative back-ends: Idefix (UKN), SQL:1999 [SIGMOD 2007]

MonetDB/XQuery has started to spread across the world already.

- X-RPC: XQuery processing in peer-to-peer networks
- XIRAF: multi-hierarchical XML documents
- Tijah: full-text retrieval for the MonetDB/XQuery system
- ~ 150 SourceForge downloads per month (MonetDB/XQuery only)

pathfinder (ˈpaːθfərdə) n. a person who makes or finds a way, esp. through unexplored areas or fields of knowledge.

Collins English Dictionary