Efficient Query Processing on Relational Data Partitioning Index Structures

SSDBM 2004
Santorini Island - Greece

Hans-Peter Kriegel, Peter Kunath, Martin Pfeifle, Matthias Renz

Database Group
Institute for Computer Science
University of Munich, Germany
Outline of the Talk

1. Introduction
2. Relational R-tree
3. Acceleration of Relational Access Methods
4. Experimental Evaluation
5. Conclusions and Future Work
Introduction

New Database Applications:
- CAD Applications
- Multimedia Information Systems
- Genome Databases
- Geo Information Systems
- ...

New Object Types:
- geographic areas
- molecular structures
- ...

→ seamless integration of spatial index structures in commercial ORDBMs needed
Outline of the Talk

1. Introduction
2. Relational R-tree
3. Acceleration of Relational Access Methods
4. Experimental Evaluation
5. Conclusions and Future Work
# Relational mapping of an R-tree directory

## Hierarchical directory

<table>
<thead>
<tr>
<th>page_id</th>
<th>page_lev</th>
<th>son_id</th>
<th>son_mbr</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROOT</td>
<td>2</td>
<td>1</td>
<td>BOX((0,0), (200,120))</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>BOX((0,0), (80,60))</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>BOX((60,20), (100,120))</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>4</td>
<td>BOX((140,120), (200,120))</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>A</td>
<td>…</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>B</td>
<td>…</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>C</td>
<td>…</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

## Relational index-table: *polygons_rtree*
Outline of the Talk

1. Introduction
2. Relational R-tree
3. Acceleration of Relational Access Methods
2. Experimental Evaluation
3. Conclusions and Future Work
General Idea

Navigational Approach

Scanning Approach

directory levels

page level

leaf level

three range scans

one extended range scan

skip tree traversal
## Navigational Approach

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n)</td>
<td>Node</td>
</tr>
<tr>
<td>(q)</td>
<td>Query</td>
</tr>
<tr>
<td>(L)</td>
<td>Level of directory node (n)</td>
</tr>
<tr>
<td>(h_B)</td>
<td>Height of the B(^+)-tree</td>
</tr>
<tr>
<td>(m)</td>
<td>Average # of index entries / node</td>
</tr>
<tr>
<td>(b)</td>
<td>Average # of index entries / block</td>
</tr>
</tbody>
</table>

### I/O and CPU cost related to a directory node \(n\) for a query \(q\)

- **I/O cost**
  \[
  \text{cost}^{\text{I/O}}_{\text{NAVIT}}(q, n) = \sum_{i=1}^{L(n)-1} m^i
  \]

- **Overall I/O-cost**
  \[
  \text{cost}^{\text{I/O}}_{\text{NAVIT}}(q, n) = c_{\text{I/O}} \cdot (h_B + m / b) \cdot \text{cnt}_{\text{NAVIT}}(q, n)
  \]

- **CPU cost**
  \[
  \text{cost}^{\text{CPU}}_{\text{NAVIT}}(q, n) = c_{\text{CPU}} \cdot m \cdot \text{cnt}_{\text{NAVIT}}(q, n)
  \]

---

Peter Kunath, University of Munich
Scanning Approach

\[ \text{I/O and CPU cost related to a directory node } n \text{ for a query } q \]

\[ cost_{SCAN}(q, n) = cost_{SCAN}^{I/O}(q, n) + cost_{SCAN}^{CPU}(q, n) \]

**overall I/O-cost:**

\[ cost_{SCAN}^{I/O}(q, n) = c_{I/O} \cdot (h_B + m^{L(n)} / b) \approx c_{I/O} \cdot (m^{L(n)} / b) \]

**overall CPU-cost:**

\[ cost_{SCAN}^{CPU}(q, n) = c_{CPU} \cdot m^{L(n)} \]
Accelerated Relational R-tree

if $cost_{SCAN}(q, n) < cost_{NAVI}(q, n) \Rightarrow$ extended range scan

Problem:
precise determination of the overlap factor $\sigma$

for the Relational R-tree:
$\sigma = \frac{V_n \cap q}{V_n}$ overlap factor

if $cost_{SCAN}^{I/O} < cost_{NAVI}^{I/O} \rightarrow$ scan

SSDBM 2004
Santorini Island Greece
Outline of the Talk

1. Introduction
2. Relational R-tree
3. Acceleration of Relational Access Methods
4. Experimental Evaluation
5. Conclusions and Future Work
Intersection Queries on the Relational R-tree

- Speedup of more than 150% for queries of low selectivity
- Improvement of more than 10,000% for highly selective queries
Outline of the Talk

1. Introduction
2. Relational R-tree
3. Acceleration of Relational Access Methods
4. Experimental Evaluation
5. Conclusions and Future Work
Conclusions and Future Work

Our Approach:

- adapts to the best of the two worlds “index” and “sequential scan”
- the optimizer can include this fine-grained approach into any query execution plan

Future Work:

- use our new approach to accelerate spatial join processing