Evaluation of Signature Files as Set Access Facilities in OODBs

Yoshiharu Ishikawa
Hiroyuki Kitagawa
Nobuo Ohbo

University of Tsukuba

May 27, 1993
Background

A research issue in OODB

Efficient indexing scheme to support

- complex data structures
- complex queries

Index techniques for complex objects

- nested index
- path index
- multiindex
Set-Valued Data

- Basic and important data structure
- Set-specific comparison operators
  ($\supseteq$, $\exists$, $\ldots$)

\[\downarrow\]

Necessity of set access facilities
Approach

Signature files as set access facilities

- Cost estimation
  - false drop probabilities
  - cost model
- Performance comparison
  - two signature file organizations (SSF, BSSF)
  - nested index
- Efficient query evaluation strategies
**Query Example** $(T \supseteq Q)$

<table>
<thead>
<tr>
<th>Student</th>
<th>id</th>
<th>name</th>
<th>hobbies</th>
<th>← target set $(T)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>s1</td>
<td>“Jeff”</td>
<td>“Baseball”</td>
<td>“Golf”</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>“Fishing”</td>
<td></td>
</tr>
<tr>
<td></td>
<td>s2</td>
<td>“Tom”</td>
<td>“Baseball”</td>
<td>“Football”</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>“Tennis”</td>
<td></td>
</tr>
<tr>
<td></td>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>

Query $Q_1$ $(T \supseteq Q)$:

```sql
SELECT name
FROM Student
WHERE hobbies \supseteq \{“Baseball”, “Fishing”\}
```

↑

query set $(Q)$

**NOTE:**

$T \ni q$ is a special case of $T \supseteq Q$
Query Example \((T \subseteq Q)\)

<table>
<thead>
<tr>
<th>Student</th>
<th>id</th>
<th>name</th>
<th>hobbies</th>
</tr>
</thead>
<tbody>
<tr>
<td>s3</td>
<td>(s3)</td>
<td>“Jim”</td>
<td>“Baseball” (\rightarrow) target set ((T)) \n</td>
</tr>
<tr>
<td>s4</td>
<td>(s4)</td>
<td>“Mike”</td>
<td>“Baseball” (\rightarrow) target set ((T)) \n</td>
</tr>
<tr>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
</tr>
</tbody>
</table>

Query \(Q_2\) \((T \subseteq Q)\):

\[
\text{select name} \\
\text{from Student} \\
\text{where hobbies} \subseteq \{\text{“Baseball”}, \text{“Fishing”}, \text{“Tennis”}\} \\
\uparrow \\
\text{query set} \((Q)\)
Organization of a Signature File

Creation of a Set Signature

<table>
<thead>
<tr>
<th>set</th>
<th>element signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>{A, B, C}</td>
<td>( (F = 8, m = 2) )</td>
</tr>
<tr>
<td>A</td>
<td>01000100</td>
</tr>
<tr>
<td>B</td>
<td>00100001</td>
</tr>
<tr>
<td>C</td>
<td>00010100</td>
</tr>
</tbody>
</table>

\[ \downarrow \text{bit-OR} \]

set signature \ 01110101

Logical Structure of a Signature File

<table>
<thead>
<tr>
<th>target set</th>
<th>target signature</th>
<th>OID</th>
</tr>
</thead>
<tbody>
<tr>
<td>{A, B, C}</td>
<td>01110101</td>
<td>s1</td>
</tr>
<tr>
<td>{A, D, E}</td>
<td>11010101</td>
<td>s2</td>
</tr>
<tr>
<td></td>
<td>\vdots</td>
<td>\vdots</td>
</tr>
</tbody>
</table>
Query Processing ($T \supseteq Q$)

<table>
<thead>
<tr>
<th>query set</th>
<th>element signature</th>
</tr>
</thead>
</table>
| $\{A, C\}$ | $A \rightarrow 01000100$  
           | $C \rightarrow 00010100$ |

$\downarrow$ bit-OR

01010100 query signature

Actual Drop

<table>
<thead>
<tr>
<th>01110101</th>
<th>s1</th>
<th>$\leftarrow {A, B, C}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>11010101</td>
<td>s2</td>
<td>$\leftarrow {A, D, E}$</td>
</tr>
<tr>
<td>10110101</td>
<td>s3</td>
<td>$\leftarrow {B, D, E}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

False Drop
**Query Processing \((T \subseteq Q)\)**

<table>
<thead>
<tr>
<th>query set</th>
<th>element signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>{A, B, C}</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>(01000100)</td>
</tr>
<tr>
<td>B</td>
<td>(00100010)</td>
</tr>
<tr>
<td>C</td>
<td>(00010100)</td>
</tr>
</tbody>
</table>

\[\Downarrow\text{bit-OR}\]

\(01110110\) query signature

<table>
<thead>
<tr>
<th>Actual Drop</th>
<th>False Drop</th>
</tr>
</thead>
<tbody>
<tr>
<td>(01010100) s1</td>
<td>(01110100) s2</td>
</tr>
<tr>
<td>(00101011) s3</td>
<td>(\vdots) s3</td>
</tr>
</tbody>
</table>

\(\leftarrow \{A, C\}\)

\(\leftarrow \{A, D\}\)

\(\leftarrow \{B, F\}\)
False Drop Probability $F_d$

Definition

$$F_d = \frac{\text{false drops}}{N - \text{actual drops}}$$

$N$: total no. of objects (signatures)

Approximate Expression

$$F_d \approx (1 - e^{-\frac{m}{F} D_t})^{m D_q} \quad (T \supseteq Q)$$
$$F_d \approx (1 - e^{-\frac{m}{F} D_q})^{m D_t} \quad (T \subseteq Q)$$

$F$: signature size (in bits)
$m$: no. of 1’s in element signature
$D_t$: cardinality of target set
$D_q$: cardinality of query set
Cost Model

Assumption

1. Retrieval cost is measured by the total of page accesses
2. Access cost of an object is 1 page
3. Cardinalities of target sets \((D_t)\) are constant \((D_t = 10\) or \(D_t = 100)\)
4. \(V\)(cardinality of the set domain) = 13,000
   \(N\)(no. of objects) = 32,000

Three Alternative Access Facilities

• Two signature file organizations:
  – SSF (Sequential Signature File)
  – BSSF (Bit-Sliced Signature File)
• Nested index (NIX)
Evaluation of Signature Files as Set Access Facilities in OODBs

SSF and BSSF

signature file       OID file

SSF

F

N

0 1 0 0 1 0 0 0
0 0 1 0 0 0 0 1
0 0 1 0 0 0 0 0
::: :::::::::
::: :::::::::
1 0 1 0 0 0 0 0

 OID file

0 1 0 1 0 0 0 0

query signature

BSSF

N

F

0 1 0 0 1 0 0 0
0 0 1 0 0 0 0 1
0 0 1 0 0 0 0 0
::: :::::::::
::: :::::::::
1 0 1 0 0 0 0 0

 OID file

bit-slice files

Yoshiharu Ishikawa, Hiroyuki Kitagawa, and Nobuo Ohbo
Retrieval Cost ($T \supseteq Q$, $D_t = 10$)

Page Accesses

\begin{align*}
\text{SSF (F = 500)} & \quad \Diamond \\
\text{SSF (F = 250)} & \quad + \\
\text{BSSF (F = 500)} & \quad \square \\
\text{BSSF (F = 250)} & \quad \times \\
\text{NIX} & \quad \triangle
\end{align*}

$D_q$ (cardinality of query set)

As $m$-values

$$m_{opt} = F \ln 2 / D_t$$

is used.

- $m_{opt} = 17 \quad (F = 250)$
- $m_{opt} = 35 \quad (F = 500)$
Small $m$-value makes

- $m_q$ small $\Rightarrow$ decrease retrieval costs
- $F_d$ large $\Rightarrow$ increase retrieval costs

$m_q$: no. of 1’s in query signature
The final qualification of the candidates is done at the false drop resolution step.

We may use only part of the $m_q$ bit-slice files.
**Smart Object Retrieval**

For BSSF \((F = 500, m = 2)\):

- \(D_q \leq 2 \Rightarrow \text{ordinal retrieval}\)
- \(D_q \geq 3 \Rightarrow \text{retrieval with only two elements}\)

**Example:**

```
select name
from Student
where hobbies \supseteq \{“Baseball”, “Fishing”, “Tennis”\}
```

\[\downarrow\]

Retrieve BSSF with

```
hobbies \supseteq \{“Baseball”, “Fishing”\}
```
Smart Retrieval Cost
$(T \supseteq Q, D_t = 10)$

Page Accesses

$\log_{10}$ Page Accesses vs. $D_q$

- BSSF $(F = 250, m = 3)$ ♦
- BSSF $(F = 250, m = 2)$ +
- BSSF $(F = 500, m = 3)$ ◊
- BSSF $(F = 500, m = 2)$ ×
- NIX △
Evaluation of Signature Files as Set Access Facilities in OODBs

Retrieval Cost \((T \subseteq Q, D_t = 10)\)

Page Accesses

\[
\begin{array}{c}
\text{SSF (m = 10) } \diamond \\
\text{SSF (m = 5) } + \\
\text{SSF (m = 2) } \square \\
\text{BSSF (m = 10) } \times \\
\text{BSSF (m = 5) } \triangle \\
\text{BSSF (m = 2) } * \\
\text{NIX } \ominus \\
\end{array}
\]
Analysis of Retrieval Cost
(BSSF, \( F = 500 \), \( m = 2 \))

For each position with value “0” in the query signature, corresponding bit-slice is retrieved.

\[
\downarrow
\]

\( D_q \) increases \( \Rightarrow \) Scan cost decreases.
Smart Retrieval Cost
\((T \subseteq Q, D_t = 10)\)
### Storage Cost

$D_t = 10$

<table>
<thead>
<tr>
<th>File</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSF ($F = 250$)</td>
<td>307</td>
</tr>
<tr>
<td>SSF ($F = 500$)</td>
<td>551</td>
</tr>
<tr>
<td>BSSF ($F = 250$)</td>
<td>313</td>
</tr>
<tr>
<td>BSSF ($F = 500$)</td>
<td>563</td>
</tr>
<tr>
<td>NIX</td>
<td>690</td>
</tr>
</tbody>
</table>

BSSF is 45% ~ 80% of NIX.

$D_t = 100$

<table>
<thead>
<tr>
<th>File</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSF ($F = 1000$)</td>
<td>1040</td>
</tr>
<tr>
<td>SSF ($F = 2500$)</td>
<td>2504</td>
</tr>
<tr>
<td>SSF ($F = 5000$)</td>
<td>4946</td>
</tr>
<tr>
<td>BSSF ($F = 1000$)</td>
<td>1063</td>
</tr>
<tr>
<td>BSSF ($F = 2500$)</td>
<td>2563</td>
</tr>
<tr>
<td>NIX</td>
<td>6531</td>
</tr>
</tbody>
</table>

BSSF is 16% ~ 28% of NIX.
### Update Cost

\( D_t = 10 \)

<table>
<thead>
<tr>
<th>File</th>
<th>Insert</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSF</td>
<td>2</td>
<td>31.5</td>
</tr>
<tr>
<td>BSSF ( F = 250 )</td>
<td>251</td>
<td>31.5</td>
</tr>
<tr>
<td>BSSF ( F = 500 )</td>
<td>501</td>
<td>31.5</td>
</tr>
<tr>
<td>NIX ( D_t = 10 )</td>
<td>30</td>
<td>30</td>
</tr>
</tbody>
</table>

\( D_t = 100 \)

<table>
<thead>
<tr>
<th>File</th>
<th>Insert</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSF</td>
<td>2</td>
<td>31.5</td>
</tr>
<tr>
<td>BSSF ( F = 1000 )</td>
<td>1001</td>
<td>31.5</td>
</tr>
<tr>
<td>BSSF ( F = 2500 )</td>
<td>2501</td>
<td>31.5</td>
</tr>
<tr>
<td>NIX ( D_t = 100 )</td>
<td>300</td>
<td>300</td>
</tr>
</tbody>
</table>

- Insert cost of BSSF is very large.
Improvement of
Insertion Cost of BSSF (1)

New Scheme

When a new page is allocated, the page is cleared with 0’s.


Only $m_t$ bit-slices files are to be updated on insertion. ($m_t$: no. of 1’s in the target signature)
### Improvement of Insertion Cost of BSSF (2)

<table>
<thead>
<tr>
<th>File</th>
<th>Insert</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSF</td>
<td>2</td>
</tr>
<tr>
<td>BSSF ($F = 250, m = 2$)</td>
<td>20</td>
</tr>
<tr>
<td>BSSF ($F = 250, m = 2$)</td>
<td>29</td>
</tr>
<tr>
<td>BSSF ($F = 500, m = 2$)</td>
<td>20</td>
</tr>
<tr>
<td>BSSF ($F = 500, m = 3$)</td>
<td>30</td>
</tr>
<tr>
<td>NIX ($D_t = 10$)</td>
<td>30</td>
</tr>
</tbody>
</table>

\[
D_t = 10
\]

<table>
<thead>
<tr>
<th>File</th>
<th>Insert</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSF</td>
<td>2</td>
</tr>
<tr>
<td>BSSF ($F = 1000, m = 2$)</td>
<td>182</td>
</tr>
<tr>
<td>BSSF ($F = 1000, m = 3$)</td>
<td>261</td>
</tr>
<tr>
<td>BSSF ($F = 2500, m = 2$)</td>
<td>193</td>
</tr>
<tr>
<td>BSSF ($F = 2500, m = 3$)</td>
<td>284</td>
</tr>
<tr>
<td>NIX ($D_t = 100$)</td>
<td>300</td>
</tr>
</tbody>
</table>

\[
D_t = 100
\]

- Insertion cost of BSSF is less than or equal to NIX
Summary (1)

Retrieval Cost (under Smart Retrieval)

- $SSF \geq BSSF$
- $T \supseteq Q$: $BSSF \sim NIX$
- $T \ni q (T \supseteq Q$ \& $D_q = 1)$:
  \[ BSSF \geq NIX \]
- $T \subseteq Q$: $BSSF \ll NIX$ (for probable $D_q$)
Summary (2)

Storage Cost

SSF, BSSF ≤ NIX

Update Cost

• Insert: SSF < NIX ≪ BSSF
• Insert (under new scheme):
  SSF < BSSF ≤ NIX
• Delete: SSF ≃ BSSF ≤ NIX


**Conclusion**

BSSFs with small m-values

$(m = 2, 3)$ are

promising set access facilities

**Future Research**

1. Cost estimation for other kinds of queries
   - $T \equiv Q$ (equality)
   - $T \cap Q$ (intersection)

2. Performance analysis in practical DB environments

3. Optimal query evaluation strategies using signature file