Decomposition of Natural Join Based on Domain-Interval Fragmented Column Indices

Elena Ivanova and Leonid Sokolinsky
South Ural State University, Chelyabinsk, Russia
ivanovaev@susu.ac.ru, Leonid.Sokolinsky@susu.ru
Architecture

SQL-server

DBMS

DB

Query

Precomputation table

Computing cluster system with many-core accelerators

DBMS coprocessor with distributed column indices
**Column Index and Domain-Interval Fragmentation**

- **The column index** is a table with two columns $A$ and $B$. The number of rows in the column index is equal to the number of rows in the indexed table. Column $B$ of index contains all the values of column $B$ in table $R$ (including duplicates). These values are sorted in ascending order inside column index.

<table>
<thead>
<tr>
<th>$R$</th>
<th>$I_{R,B}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>$B$</td>
</tr>
<tr>
<td>0</td>
<td>36</td>
</tr>
<tr>
<td>1</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>36</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>74</td>
</tr>
<tr>
<td>5</td>
<td>27</td>
</tr>
<tr>
<td>6</td>
<td>58</td>
</tr>
</tbody>
</table>

$\mathbb{D}_B$ is the domain of attribute $B$

Fragsments of index: $I^0_{R,B}$, $I^1_{R,B}$, ..., $I^{k-1}_{R,B}$
Example of Fragmented Column Index

A fragment is \[
\begin{cases}
0, & \text{if } x \cdot B \in [0,7) \\
1, & \text{if } x \cdot B \in [7,12]
\end{cases}
\]
Decomposition of Natural Join Operation $R \bowtie S$

A fragment is 

$$f(x. B) = \begin{cases} 
0, & \text{if } x. B \in [0, 7) \\
1, & \text{if } x. B \in [7, 12] 
\end{cases}$$

<table>
<thead>
<tr>
<th>$R$</th>
<th>$S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>$A$</td>
</tr>
<tr>
<td>$B$</td>
<td>$B$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Column indices for attributes $B$ of $R$ and $S$
Independent Join of Fragments

Node 0

\[ I^{0}_{R.B} \]

\[
\begin{array}{cc}
A & B \\
3 & 1 \\
2 & 5 \\
\end{array}
\]

\[ I^{0}_{S.B} \]

\[
\begin{array}{cc}
A & B \\
2 & 3 \\
4 & 3 \\
0 & 5 \\
\end{array}
\]

\[ p^{0} \]

\[
\begin{array}{cc}
R.A & S.A \\
2 & 0 \\
\end{array}
\]

Node 1

\[ I^{1}_{R.B} \]

\[
\begin{array}{cc}
A & B \\
0 & 10 \\
1 & 12 \\
\end{array}
\]

\[ I^{1}_{S.B} \]

\[
\begin{array}{cc}
A & B \\
3 & 10 \\
1 & 11 \\
\end{array}
\]

\[ p^{1} \]

\[
\begin{array}{cc}
R.A & S.A \\
0 & 3 \\
\end{array}
\]
Passing of Partial Result to SQL-server

Node 0

\[
\begin{array}{cc}
R.A & S.A \\
2 & 0 \\
\end{array}
\]

Node-coordinator

\[
\begin{array}{cc}
R.A & S.A \\
2 & 0 \\
0 & 3 \\
\end{array}
\]

Node 1

\[
\begin{array}{cc}
R.A & S.A \\
0 & 3 \\
\end{array}
\]

\[
\begin{array}{cc}
R.A & S.A \\
2 & 0 \\
0 & 3 \\
\end{array}
\]

\[
P
\]

\[
\begin{array}{cc}
R.A & S.A \\
2 & 0 \\
0 & 3 \\
\end{array}
\]

P is precomputation table
Join on SQL-server by precomputation table

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>Ni</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>Au</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>Pb</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>Ag</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>Pb</td>
</tr>
<tr>
<td>1</td>
<td>11</td>
<td>Pb</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>Ni</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>Fr</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>Ag</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>R.A</td>
<td>S.A</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R.B</th>
<th>R.C</th>
<th>S.D</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Pb</td>
<td>Pb</td>
</tr>
<tr>
<td>10</td>
<td>Ni</td>
<td>Fr</td>
</tr>
</tbody>
</table>
Prototype of DBMS coprocessor

- Prototype work on one Intel Xeon Phi (60 cores per 1.1 MHz) in native mode.
- All column index fragments are stored in main memory in compressed form.

Number of fragments: 4
Number of threads: 2
Operation: R⨝S
Prototype of DBMS coprocessor

- Prototype work on one Intel Xeon Phi (60 cores per 1.1 MHz) in native mode.
- All column index fragments are stored in main memory in compressed form.
Prototype of DBMS coprocessor

- Prototype work on one Intel Xeon Phi (60 cores per 1.1 MHz) in native mode.
- All column index fragments are stored in main memory in compressed form.

Number of fragments: 4
Number of threads: 2
Operation: R⨝S
Performance Evaluation

- $R \bowtie S$
- Number of tuples $I_{R,B}$ is 600 000
- Number of tuples $I_{S,B}$ is 60 000 000
Conclusion

- We presented a decomposition of the natural join operator based on the column indices and the domain-interval fragmentation.
- Our approach was evaluated using the prototype DBMS coprocessor. Experiments showed its efficiency for a resource-intensive natural join operator.