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

Net

Node 0

$P_1 \ldots P_u$

$M_0$

Node $k-1$

$P_1 \ldots P_u$

$M_{k-1}$

DBMS coprocessor with distributed column indices
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.

$\mathcal{D}_B$ is the domain of attribute $B$.
Example of Fragmented Column Index

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

A fragment is
\[
\begin{cases} 
0, \text{ if } x.B \in [0,7) \\
1, \text{ if } x.B \in [7,12]
\end{cases}
\]

<table>
<thead>
<tr>
<th>Column indices for attributes $B$ of $R$ and $S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I^0_{R.B}$</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

$R$

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>Ni</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>Au</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>Pb</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>Ag</td>
<td></td>
</tr>
</tbody>
</table>

$S$

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

**Node 0**

\[
\begin{array}{c|c}
I^0_{R.B} & I^0_{S.B} \\
\hline
A & B \\
3 & 1 \\
2 & 5 \\
\end{array}
\]

\[
\begin{array}{c|c}
p^0 \\
R.A & S.A \\
2 & 0 \\
\end{array}
\]

**Node 1**

\[
\begin{array}{c|c}
I^1_{R.B} & I^1_{S.B} \\
\hline
A & B \\
0 & 10 \\
1 & 12 \\
\end{array}
\]

\[
\begin{array}{c|c}
p^1 \\
R.A & S.A \\
0 & 3 \\
\end{array}
\]

---

25-29 May, 2015  DC VIS - Distributed Computing, Visualization and Biomedical Engineering  www.mipro.hr
Passing of Partial Result to SQL-server

Node 0

\[ P^0 \]

\[
\begin{array}{|c|c|}
\hline
R.A & S.A \\
\hline
2 & 0 \\
\hline
\end{array}
\]

Node-coordinator

\[
\begin{array}{|c|c|}
\hline
R.A & S.A \\
\hline
2 & 0 \\
\hline
\end{array} \quad \begin{array}{|c|c|}
\hline
R.A & S.A \\
\hline
0 & 3 \\
\hline
\end{array}
\]

Node 1

\[ P^1 \]

\[
\begin{array}{|c|c|}
\hline
R.A & S.A \\
\hline
0 & 3 \\
\hline
\end{array}
\]

\[ P \]

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

\[ P \text{ is precomputation table} \]
Join on SQL-server by precomputation table

\[ R \]

\[
\begin{array}{ccc}
A & B & C \\
0 & 10 & Ni \\
1 & 10 & Au \\
2 & 5 & Pb \\
3 & 1 & Ag \\
\end{array}
\]

\[ S \]

\[
\begin{array}{ccc}
A & B & D \\
0 & 5 & Pb \\
1 & 11 & Pb \\
2 & 3 & Ni \\
3 & 10 & Fr \\
4 & 2 & Ag \\
\end{array}
\]

\[ P \]

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

\[ \]

\[
\begin{array}{ccc}
R.B & R.C & S.D \\
5 & Pb & Pb \\
10 & Ni & Fr \\
\end{array}
\]
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: $R \bowtie S$
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.