Direct Access for Conjunctive Queries with Aggregation

Nofar Carmeli
Example

Goal: get a sense of how many views come from a contributor

<table>
<thead>
<tr>
<th>Contributor</th>
<th>Resource</th>
<th>Date</th>
<th>Views</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>CS101</td>
<td>01/01/23</td>
<td>4</td>
</tr>
<tr>
<td>Alice</td>
<td>CS101</td>
<td>02/01/23</td>
<td>125</td>
</tr>
<tr>
<td>Bob</td>
<td>CS101</td>
<td>01/01/23</td>
<td>4</td>
</tr>
<tr>
<td>Bob</td>
<td>CS101</td>
<td>02/01/23</td>
<td>125</td>
</tr>
<tr>
<td>Alice</td>
<td>Sophrology</td>
<td>01/01/23</td>
<td>26</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
<th>Date</th>
<th>Views</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS101</td>
<td>01/01/23</td>
<td>4</td>
</tr>
<tr>
<td>CS101</td>
<td>02/01/23</td>
<td>125</td>
</tr>
<tr>
<td>Sophrology</td>
<td>01/01/23</td>
<td>26</td>
</tr>
</tbody>
</table>

Q(sum(views), contributor) ← content(contributor, resource), activity(resource, date, views)

1. Join

2. Group by Contributor

3. Sum

4. Sort by Views

<table>
<thead>
<tr>
<th>Contributor</th>
<th>Views</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>155</td>
</tr>
<tr>
<td>Bob</td>
<td>129</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Views</th>
<th>Contributor</th>
</tr>
</thead>
<tbody>
<tr>
<td>129</td>
<td>Bob</td>
</tr>
<tr>
<td>155</td>
<td>Alice</td>
</tr>
</tbody>
</table>
Example

Goal: get a sense of how many views come from a contributor

4. Sort by Views

<table>
<thead>
<tr>
<th>Views</th>
<th>Contributor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Eve</td>
</tr>
<tr>
<td>103</td>
<td>Frank</td>
</tr>
<tr>
<td>117</td>
<td>Dave</td>
</tr>
<tr>
<td>129</td>
<td>Bob</td>
</tr>
<tr>
<td>136</td>
<td>Carol</td>
</tr>
<tr>
<td>155</td>
<td>Alice</td>
</tr>
<tr>
<td>304</td>
<td>George</td>
</tr>
</tbody>
</table>

5. Get statistics

A) Median

129 Bob

C) Histogram

[0,50) [50,100) [100,150) [150,200) [200,250) [250,300) [300,350)
Definition: Ranked Direct Access

- Simulate a sorted array containing the answers
- Given i, returns the i\textsuperscript{th} answer or “out of bound”.
- Ranked: user-specified order

**Standard approach**

**Our approach**
Overview of Tasks

- ranked access
- ranked pagination
- quantile computation
- histogram computation

"The rows skipped by an OFFSET clause still have to be computed inside the server; therefore a large OFFSET might be inefficient."

www.postgresql.org
Overview of Tasks

- ranked access
- ranked pagination
- ranked enumeration
- random-ordered enumeration
- top k

- direct access
- quantile computation
- histogram computation
- counting
- sampling

[C, Zeevi, Berkholz, Kimelfeld, Schweikardt; PODS 20]
Overview of Tasks

- ranked access
- direct access
- ranked enumeration
- random-ordered enumeration
- enumeration
- sampling
- evaluation
- quantile computation
- histogram computation
- counting
- ranked pagination
- top k
Research question

When can we do ranked access with (quasi)linear preprocessing and log access time?

Our focus: conjunctive queries with aggregation, lexicographic orders
Plan

• Motivation
• Dichotomy without aggregation
• Aggregation not affecting the order
  • Using annotations, the dichotomy still holds
• Aggregation affecting the order
  • Limited tractability using general annotations
  • Local annotations
    • In some cases (full query or idempotent semiring), equivalent to hardness of CQs with FDs
• Conclusion
Given: conjunctive query $Q$, ordering $L$ of $\text{free}(Q)$,

**lexicographic access in $\langle \loglinear, \log \rangle$**

↑*

acyclic free-connex, no disruptive trio

* Lower bound requires:

**sHyperclique hypothesis:** $\forall k \geq 3$ the existence of a $k$-hyperclique in a $(k - 1)$-uniform hypergraph cannot be decided in quasilinear time in the number of edges

**sBMM hypothesis:** Boolean matrices cannot be multiplied in quasilinear time in the number of the 1 entries
Definition: Free-Connex Acyclic

An acyclic CQ has a graph with:
A free-connex CQ also requires:

1. A node for every atom
2. Tree
3. For every variable: the nodes containing it form a subtree
4. Remains acyclic when introducing an atom with the free variables

\[ Q(x, y, z) \leftarrow R_1(x, y), R_2(y, z), R_3(z, w) \]
Given: conjunctive query $Q$, ordering $L$ of free($Q$), lexicographic access in $\langle \loglinear, \log \rangle$ \uparrow^* acyclic free-connex, no disruptive trio

Disruptive Trio Definition

$v_1 \xrightarrow{X} v_2$
$v_3$

share an atom

last out of the three

Examples

$Q_1(v_1, v_2, u) \leftarrow R(v_1, u), S(u, v_2)$
$Q_2(u, v_1, v_2) \leftarrow R(v_1, u), S(u, v_2)$

* Lower bound requires:

sHyperclique hypothesis: $\forall k \geq 3$ the existence of a $k$-hyperclique in a $(k - 1)$-uniform hypergraph cannot be decided in quasilinear time in the number of edges

sBMM hypothesis: Boolean matrices cannot be multiplied in quasilinear time in the number of the 1 entries
Plan

• Motivation
• Dichotomy without aggregation
• Aggregation not affecting the order
  • Using annotations, the dichotomy still holds
• Aggregation affecting the order
  • Limited tractability using general annotations
  • Local annotations
    • In some cases (full query or idempotent semiring), equivalent to hardness of CQs with FDs
• Conclusion
Aggregation not affecting the order

- Approach: translate aggregates to semiring annotations.
- Example:

\[
Q(\mathit{x}, \mathit{y}, \text{sum}(\mathit{w})) \leftarrow R(\mathit{x}, \mathit{w}), S(\mathit{y})
\]

1) translate

\[
R \quad S
\]

\[
\begin{array}{cc}
\mathit{x} & \mathit{w} \\
\mathit{x}_1 & 2 \\
\mathit{x}_1 & 5 \\
\mathit{x}_2 & 8 \\
\end{array}
\]

\[
\begin{array}{c}
\mathit{y} \\
\mathit{y}_1 \\
\mathit{y}_2 \\
\end{array}
\]

\[
Q(\mathit{x}, \mathit{y}, \mathit{w}) \leftarrow R(\mathit{x}, \mathit{w}), S(\mathit{y})
\]

\[
\begin{array}{cc}
\mathit{x} & \mathit{w} \\
\mathit{x}_1 & 2 \\
\mathit{x}_1 & 5 \\
\mathit{x}_2 & 8 \\
\end{array}
\]

\[
\begin{array}{c}
\mathit{y} \\
\mathit{y}_1 \\
\mathit{y}_2 \\
\end{array}
\]

\[
Q(\mathit{x}, \mathit{y}, \star) \leftarrow R(\mathit{x}, \mathit{w}), S(\mathit{y})
\]

\[
\begin{array}{c}
\mathit{y} \\
\mathit{y}_1 \\
\mathit{y}_2 \\
\end{array}
\]

2) handle projections

3) Use access algorithm for CQs

\[
\text{The } 2^{\text{nd}} \text{ answer is: } \begin{array}{cc}
\mathit{x}_1 & \mathit{y}_2 \\
\end{array}
\]

4) multiply annotations

\[
\text{The } 2^{\text{nd}} \text{ answer is: } \begin{array}{cc}
\mathit{x}_1 & \mathit{y}_2 \\
\end{array} \cdot (2 + 5) \cdot 1
\]

answers

\[
\begin{array}{ccc}
\mathit{x} & \mathit{y} & \text{sum}(\mathit{w}) \\
\mathit{x}_1 & \mathit{y}_1 & 2 + 5 \\
\mathit{x}_1 & \mathit{y}_2 & 2 + 5 \\
\mathit{x}_2 & \mathit{y}_1 & 8 \\
\mathit{x}_2 & \mathit{y}_2 & 8 \\
\end{array}
\]

answers

\[
\begin{array}{cc}
\mathit{x} & \mathit{y} \\
\mathit{x}_1 & \mathit{y}_1 \\
\mathit{x}_1 & \mathit{y}_2 \\
\mathit{x}_2 & \mathit{y}_1 \\
\mathit{x}_2 & \mathit{y}_2 \\
\end{array}
\]

\[
\begin{array}{cc}
(2 + 5) \cdot 1 \\
(2 + 5) \cdot 1 \\
8 \cdot 1 \\
8 \cdot 1 \\
\end{array}
\]
Given: CQ* $Q(\vec{x}, \ast)$

**lexicographic access in** $\langle \loglinear, \log \rangle$

$\uparrow \ast$

**acyclic free-connex, no disruptive trio**

* Lower bound requires:

**sHyperclique hypothesis:** $\forall k \geq 3$ the existence of a $k$-hyperclique in a $(k - 1)$-uniform hypergraph cannot be decided in quasilinear time in the number of edges

**sBMM hypothesis:** Boolean matrices cannot be multiplied in quasilinear time in the number of the 1 entries
Using Log-time Commutative Semirings

- Commutative semiring: \((\mathcal{K}, \oplus, \otimes, \overline{0}, \overline{1})\)
  - \(\mathcal{K}\) is a domain of elements
  - \((\mathcal{K}, \oplus, \overline{0})\) is a commutative monoid ("addition")
    - \((a \oplus b) \oplus c = a \oplus (b \oplus c)\) (associative)
    - \(a \oplus b = b \oplus a\) (commutative)
    - \(a \oplus \overline{0} = a\) (\(\overline{0}\) neutral)
  - \((\mathcal{K}, \otimes, \overline{1})\) is a commutative monoid ("multiplication")
    - \(a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)\) (distributive)
    - \(a \otimes \overline{0} = \overline{0}\)

- In databases [Green, Karvounarakis, Tannen 2007]:
  - Each tuple is annotated with a semiring element
  - When joining tuples, multiply the annotations
  - When projecting, sum up the group’s annotation
Aggregations and Semirings

• Using log-time commutative semirings:
  • Sum: numerical semiring \((\mathbb{Q}, +, \cdot, 0, 1)\)
  • Count: counting semiring \((\mathbb{N}, +, \cdot, 0, 1)\)
  • Min: min-tropical semiring \((\mathbb{Q} \cup \{\infty\}, \min, +, \infty, 0)\)
  • Max: max-tropical semiring \((\mathbb{Q} \cup \{-\infty\}, \max, +, -\infty, 0)\)

• Average:
  • combine sum and count

• Count-Distinct:
  • No semiring translation
  • Harder than the others
    • \(Q(x, \text{distinct}(z)) \leftarrow R(x, y), S(y, z)\) hard (assuming small-universe hitting set conjecture)
  • In case of log-size domain: use set semiring \((2^\Omega, \cup, \cap, \emptyset, \Omega)\)

Small-universe Hitting Set Conjecture [Williams 15]:
Given two sets \(U\) and \(V\) of size \(N\), each containing sets over \{1, 2, ..., \(d\)\}, does \(U\) contains a set that shares an element with every set in \(V\)? Conjecture: it takes \(N^{2 - o(1)}\) time for every function \(d = \omega(\log N)\).
Plan

• Motivation
• Dichotomy without aggregation
• Aggregation not affecting the order
  • Using annotations, the dichotomy still holds
• Aggregation affecting the order
  • Limited tractability using general annotations
  • Local annotations
    • In some cases (full query or idempotent semiring), equivalent to hardness of CQs with FDs
• Conclusion
Incorporating Aggregation in the Order

• Examples:
  • $Q_1(x, y, \ast) \leftarrow R(x), S(y)$ easy (from dichotomy)
  • $Q_2(\ast, x, y) \leftarrow R(x), S(y)$ hard (assuming 3SUM)
  • $Q_3(x, \ast, y) \leftarrow R(x), S(y)$ easy (from sufficient condition)

Sufficient condition:
Consider a CQ $Q(\vec{x}, \ast, \vec{z})$.
If every atom contains either all of $\vec{z}$ or none of $\vec{z}$, and $Q'(\vec{x}, \vec{z})$ is acyclic free-connex with no disruptive trio, then* lexicographic access in $<$loglinear,log$>$ for $Q(\vec{x}, \ast, \vec{z})$.

* Assuming $\otimes$-monotonicity.

$\otimes$-monotonicity:
for every $c$, either $c \otimes a \leq c \otimes b$ whenever $a \leq b$, or $c \otimes b \leq c \otimes a$ whenever $a \leq b$.

3SUM Conjecture:
given a set of $N$ elements from $\{-N^3, \ldots, N^3\}$, are there distinct elements $a, b, c$ such that $a + b = c$? Conjecture: it takes $N^{2-o(1)}$ time.
Incorporating Aggregation in the Order

• Examples:
  - $Q_1(x, y, \star) \leftarrow R(x), S(y)$ easy (from dichotomy)
  - $Q_2(\star, x, y) \leftarrow R(x), S(y)$ hard (assuming 3SUM)
  - $Q_3(x, \star, y) \leftarrow R(x), S(y)$ easy (from sufficient condition)
  - $Q_4(\text{sum}(w), x, y) \leftarrow R(x, w), S(y)$ easy (locally annotated)
    - Translated to the hard $Q_2(\star, x, y) \leftarrow R(x), S(y)$
    - However, diverse annotation only in $R$
    - Equivalent in hardness to the easy $Q_4'(z, x, y) \leftarrow R(x, z), S(y)$ with the FD $x \rightarrow z$

Use FDs for more tractable cases

[Kimelfeld, Riedewald; TODS 23]
Incorporating Aggregation in the Order

• Examples:
  1. $Q_1(x, y, \star) \leftarrow R(x), S(y)$ easy (from dichotomy)
  2. $Q_2(\star, x, y) \leftarrow R(x), S(y)$ hard (assuming 3SUM)
  3. $Q_3(x, \star, y) \leftarrow R(x), S(y)$ easy (from sufficient condition)
  4. $Q_4(\text{sum}(w), x, y) \leftarrow R(x, w), S(y)$ easy (locally annotated)
     - Translated to the hard $Q_2(\star, x, y) \leftarrow R(x), S(y)$
     - However, diverse annotation only in $R$
     - Equivalent in hardness to the easy $Q_4'(z, x, y) \leftarrow R(x, z), S(y)$ with the FD $x \rightarrow z$

Use FDs for more tractable cases [C, Tziavelis, Gatterbauer, Kimelfeld, Riedewald; TODS 23]

Full classification for local annotations in self-join-free case of:
- full CQ$\star$ or $\oplus$-idempotent semiring

$\oplus$-idempotent: for every $a$, $a \oplus a = a$. 

Min
Max
Distinct (log domain)
Incorporating Aggregation in the Order

Examples:

• $Q_1(x, y, \star) \leftarrow R(x), S(y)$ easy (from dichotomy)
• $Q_2(\star, x, y) \leftarrow R(x), S(y)$ hard (assuming 3SUM)
• $Q_3(x, \star, y) \leftarrow R(x), S(y)$ easy (from sufficient condition)
• $Q_4(sum(w), x, y) \leftarrow R(x, w), S(y)$ easy (locally annotated)
  • Translated to the hard $Q_2(\star, x, y) \leftarrow R(x), S(y)$
  • However, diverse annotation only in $R$
  • Equivalent in hardness to the easy $Q_4'(z, x, y) \leftarrow R(x, z), S(y)$ with the FD $x \rightarrow z$
• $Q_5(count(), x, y) \leftarrow R(x, w), S(y, z)$ easy (ad-hoc algorithm)

\[\begin{array}{c|c|c|}
| R & S & \text{translate} \\
|---|---|---|
| & | \\
| x & w & \\
| $x_1$ & $w_1$ & \\
| $x_1$ & $w_2$ & \\
| $x_2$ & $w_1$ & \\
| & & \\
| y & z & \\
| $y_1$ & $z_1$ & \\
| $y_1$ & $z_2$ & \\
| $y_1$ & $z_3$ & \\
\end{array}\]

$Q_5(count(), x, y) \leftarrow R(x, w), S(y, z)$

$Q_2(\star, x, y) \leftarrow R(x), S(y)$

\[\begin{array}{c|c|c|}
| R & S & \\
|---|---|---|
| & | \\
| x & \\
| $x_1$ & 2 \\
| $x_2$ & 1 \\
| & | \\
| y & \\
| $y_1$ & 3 \\
\end{array}\]

$Q_5(count(), x, y) \leftarrow R(x, w), S(y, z)$

$Q_2(\star, x, y) \leftarrow R(x), S(y)$

\[\begin{array}{c|c|c|}
| R & S & \\
|---|---|---|
| & | \\
| x & \\
| $x_1$ & 2 \\
| $x_2$ & 1 \\
| & | \\
| y & \\
| $y_1$ & 3 \\
\end{array}\]

$Q_5(count(), x, y) \leftarrow R(x, w), S(y, z)$

$Q_2(\star, x, y) \leftarrow R(x), S(y)$

\[\begin{array}{c|c|c|}
| R & S & \\
|---|---|---|
| & | \\
| x & \\
| $x_1$ & 2 \\
| $x_2$ & 1 \\
| & | \\
| y & \\
| $y_1$ & 3 \\
\end{array}\]

$Q_5(count(), x, y) \leftarrow R(x, w), S(y, z)$

$Q_2(\star, x, y) \leftarrow R(x), S(y)$
Conclusion

• Summary
  • Motivation
  • Dichotomy without aggregation
  • Aggregation not affecting the order
    • Using annotations, the dichotomy still holds
  • Aggregation affecting the order
    • Limited tractability using general annotations
    • Local annotations
      • In some cases (full query or idempotent semiring), equivalent to hardness of CQs with FDs

• Outlook
  • Open cases
  • Self-Joins
  • Time requirements for hard cases
    • Known for join queries [Bringmann, C, Mengel; PODS 22]
  • More complicated settings
    • Other orders
    • Other queries
    • Supporting updates