Single-Round Multi-Join Evaluation

Bas Ketsman
Outline

1. Introduction

2. Parallel-Correctness

3. Transferability

4. Special Cases
Motivation

Single-round Multi-joins

- Less rounds / barriers

Formal framework for reasoning about distributed query evaluation and optimization
1-Round MPC model

Query $Q$

Global instance: $I$

Data partitioning

Local instances:

- $I_1$
- $I_2$
- $I_3$

Local outputs:

- $Q(I_1)$
- $Q(I_2)$
- $Q(I_3)$

Global output: $Q(I_1) \cup Q(I_2) \cup Q(I_3)$

Modeled by a partitioning policy $P$

[ Koutris & Suciu 2011 ]
Main Questions: Question 1

Given target query and a distribution policy:

Does the simple algorithm work?

- **Parallel-Correctness**

"Is query parallel-correct for current distribution policy?"

- **If yes:**
  
  No data reshuffling needed!

- **If no:**

  Choose one that works and reshuffle.

**future work**: Which one is cheapest to obtain?
Main Questions: Question 2

It may be unpractical to reason about distribution policies
- Sometimes complex to reason about
- May be hidden behind abstraction layer
- May not have been chosen yet

Given target query and previously computed query:

Do we need to reshuffle?

Parallel-Correctness  Transferability

“Given $Q_1, Q_2$: in which order to compute?”

▶ If transferability from $Q_1$ to $Q_2$:
    Compute $Q_1$ first, then $Q_2$ for free!
Outline

1. Introduction

2. Parallel-Correctness

3. Transferability

4. Special Cases
Distribution Policies

**Network** \( \mathcal{N} \) is a finite set of machines

\[ [\text{Zinn et al. 2013}] \]

**Definition**

A *distribution policy* \( P \) is a total function mapping facts (over \( \text{dom} \)) to sets of machines in \( \mathcal{N} \)

- Based on granularity of facts
- No context
- Obtainable in distributed fashion
Distribution Policies

Network $\mathcal{N}$ is a finite set of machines

Instance $I = \{R(a,b), R(b,a), S(a)\}$

$\textit{distribution of } I \text{ based on } P$
Example Policy: Hypercube

[Afrati & Ullman 2010, Beame, Koutris & Suciu 2014]

\[(x, y, z) \leftarrow R(x, y), S(y, z), T(z, x)\]

Partitioning of complete valuations over machines in instance independent way through hashing of domain values
Simple Evaluation Algorithm

Global instance: $I$

Data partitioning

Local instances:
- $I_1$
- $I_2$
- $I_3$

Local outputs:
- $Q(I_1)$
- $Q(I_2)$
- $Q(I_3)$

Global output:
$Q(I_1) \cup Q(I_2) \cup Q(I_3)$

Notation

$$[Q, P](I) = \bigcup_{\kappa \in \mathcal{N}} Q(\text{dist}_{P,I}(\kappa))$$
Parallel-Correctness

Definition

$Q$ is \textit{parallel-correct on $I$ w.r.t. $P$}, iff $[Q, P](I) = Q(I)$

Definition (w.r.t. all instances)

$Q$ is \textit{parallel-correct w.r.t. $P$} iff $Q$ is parallel-correct w.r.t. $P$ on every $I$
Conjunctive Queries

Conjunctive Query:
Existentially quantified conjunction of relational atoms

\[ T(\bar{x}) \leftarrow R_1(\bar{y}_1), \ldots, R_m(\bar{y}_m) \]

Valuations: \( V = \) mapping from variables to domain elements
If \( V(\text{body}_Q) \subseteq I \) then output \( V(\text{head}_Q) \).

CQs are monotone (\( Q(I) \subseteq Q(I \cup J) \forall I, J \)):
- CQs are parallel-sound on every \( P \)
- parallel-correct iff parallel-complete

\[ [Q, P](I) = Q(I), \forall I \text{ iff } Q(I) \subseteq [Q, P](I), \forall I \]
Parallel-Correctness
Sufficient Condition

(\text{PC0}) \text{ for every valuation } V \text{ for } Q,

\[ \bigcap_{f \in V(\text{body}_Q)} P(f) \neq \emptyset. \]

\textbf{Intuition:} Facts required by a valuation meet at some machine

\textbf{Lemma} \hspace{1cm} (\text{PC0}) \text{ implies } Q \text{ parallel-correct w.r.t. } P.

Not necessary
(PC0) not Necessary

Example

Distribution policy $P$

Query $Q$: $T(x, z) \leftarrow R(x, y), R(y, z), R(x, x)$

$V = \{x, z \rightarrow a, y \rightarrow b\}$

Requires:

$R(a, b) \quad R(b, a) \quad R(a, a)$

Derives: Do not meet

$T(a, a)$

$V' = \{x, y, z \rightarrow a\}$

Requires:

$\supseteq R(a, a)$

Derives:

$T(a, a)$
Parallel-Correctness
Characterization

Lemma

$Q$ is parallel-correct w.r.t. $P$ iff

(\text{PC1}) \text{ for every } \textit{minimal} \text{ valuation } V \text{ for } Q,$

$$\bigcap_{f \in V(\text{body}_Q)} P(f) \neq \emptyset.$$ 

Definition

$V$ is \textit{minimal} if no $V'$ exists, where

$V'(\text{head}_Q) = V(\text{head}_Q), \ V'(\text{body}_Q) \subsetneq V(\text{body}_Q).$
Parallel-Correctness
Example

Query $Q$: $T(x, z) \leftarrow R(x, y), R(y, z), R(x, x)$

$V = \{x, z \rightarrow a, y \rightarrow b\}$

Requires:

\[
\begin{array}{ccc}
R(a, b) & R(b, a) & R(a, a) \\
\end{array}
\]

Derives:

\[
\begin{array}{c}
T(a, a) \\
\end{array}
\]

$V' = \{x, y, z \rightarrow a\}$

Requires:

\[
\begin{array}{c}
R(a, a) \\
\end{array}
\]

Derives:

\[
\begin{array}{c}
T(a, a) \\
\end{array}
\]

Notice: $Q$ is minimal CQ

CQ is minimal iff injective valuations are minimal

Proposition

Testing whether a valuation is minimal is coNP-complete.
Theorem

Deciding whether $Q$ is parallel-correct w.r.t. $P$ is $\Pi_2^P$-complete.

Proof:

- Lower bound: Reduction from $\Pi_2$-QBF
- Upper bound: (PC1) but, requires proper formalization of $P$
Parallel-Correctness: Complexity

<table>
<thead>
<tr>
<th></th>
<th>CQ</th>
<th>⋯</th>
<th>CQ{\neq, \cup}</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{P}_{\text{fin}}$</td>
<td>$\Pi^p_2$-c</td>
<td>$\Pi^p_2$-c</td>
<td>$\Pi^p_2$-c</td>
</tr>
<tr>
<td>$\mathcal{P}_{\text{enum}}$</td>
<td>$\Pi^p_2$-c</td>
<td>$\Pi^p_2$-c</td>
<td>$\Pi^p_2$-c</td>
</tr>
<tr>
<td>$\mathcal{P}^k_{\text{nondet}}$</td>
<td>$\Pi^p_2$-c</td>
<td>$\Pi^p_2$-c</td>
<td>$\Pi^p_2$-c</td>
</tr>
</tbody>
</table>

Robust under adding inequalities and union

Inequalities:

$$T(\bar{x}) \leftarrow R_1(\bar{y}_1), \ldots, R_m(\bar{y}_m), x \neq y, y \neq z$$

Union:

$$\mathcal{Q} = \{\mathcal{Q}_1, \ldots, \mathcal{Q}_k\}$$

with head$_{\mathcal{Q}_1}, \ldots, $head$_{\mathcal{Q}_k}$ over same relation.
Safe Negation

\[ T(\bar{x}) \leftarrow R_1(\bar{y}_1), \ldots, R_m(\bar{y}_m), \neg S_1(\bar{z}_1), \ldots, \neg S_k(\bar{z}_k) \]

with \( \text{vars}(\text{neg}_Q) \subseteq \text{vars}(\text{pos}_Q) \).

In general:

<table>
<thead>
<tr>
<th>( \mathcal{P}_{\text{enum}} )</th>
<th>( {\neg} )</th>
<th>( \cdots )</th>
<th>( {\neg, \cup, \neq} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>coNEXP-c</td>
<td>coNEXP-c</td>
<td></td>
<td></td>
</tr>
<tr>
<td>coNEXP-c</td>
<td>coNEXP-c</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Surprisingly we found this via CQ\(^-\) containment!!
We thought $\Pi^2_p$ completeness of $\text{CQ}^-$ containment was folklore

**Theorem**

In general, containment for $\text{CQ}^-$ is coNEXPTIME-complete

**Proof:**
- Lower bound: succinct 3-colorability
- Upper bound: guess instances over bounded domain
1. Introduction

2. Parallel-Correctness

3. Transferability

4. Special Cases
Computing Multiple Queries

\[ Q \rightarrow \]

\[ Q(I) \leftarrow \]

\[ Q' \rightarrow \]

\[ Q'(I) \leftarrow \]

\[ I \]

Redistribution
Computing Multiple Queries

When can $Q'$ be evaluated on data partitioning used for $Q$?

No reshuffling
Transferability

**Definition**

\[ Q \rightarrow_T Q' \text{ iff } Q' \text{ is parallel-correct on every } P \text{ where } Q \text{ is parallel-correct on } \]

**Example**

\[ Q : T() \leftarrow R(x, y), R(y, z), R(z, w) \]

\[ Q' : N() \leftarrow R(x, y), R(y, x) \]

\[ Q \rightarrow_T Q' \]
Lemma

\[ Q \rightarrow_T Q' \text{ iff} \]

(C2) for every minimal valuation \( V' \) for \( Q' \) there is a minimal valuation \( V \) for \( Q \), s.t.

\[ V'(\text{body}_{Q'}) \subseteq V(\text{body}_Q). \]
Lemma

\( Q \rightarrow_T Q' \) iff

(C2) for every minimal valuation \( V' \) for \( Q' \) there is a minimal valuation \( V \) for \( Q \), s.t.

\[ V'(\text{body}_{Q'}) \subseteq V(\text{body}_{Q}). \]

Theorem

Deciding \( Q \rightarrow_T Q' \) is \( \Pi_3^P \)-complete.

- Lower bound: Reduction from \( \Pi_3 \)-QBF
- Upper bound: Characterization

Based on query structure alone, not on distribution policies
Outline

1. Introduction
2. Parallel-Correctness
3. Transferability
4. Special Cases
Hypercube

Algorithm:

- Reshuffling based on structure of $Q$

$\mathcal{H}(Q) = \text{family of Hypercube policies for } Q.$

Definition

$Q \rightarrow_{H} Q'$ iff $Q'$ is parallel-correct w.r.t. every $P \in \mathcal{H}(Q).$
Hypercube

Two properties:

- **Q-generous:** for every valuation facts meet on some machine ($\forall P \in H(Q)$)
- **Q-scattered:** there is a policy scattering facts in such a way that no facts meet by coincidence ($\forall I$)

**Theorem**

Deciding whether $Q \rightarrow^H Q'$ is NP-complete

(also when $Q$ or $Q'$ is acyclic)
Tractable results **future work**

- Queries classes
- Concrete families of distribution policies
  (some other special cases in [AGKNS 2011])

Hybrid techniques / Tradeoffs **future work**

- Single-round Multi-join vs multi-rounds?
- Combining queries vs sequential distributed evaluation?
Joint work with

Tom Ameloot, Gaetano Geck, Frank Neven and Thomas Schwentick

▶ Parallel-Correctness and Transferability for Conjunctive Queries, PODS 2015.
▶ Parallel-Correctness and Containment for Conjunctive Queries with Union and Negation, ICDT 2016.
▶ Data partitioning for single-round multi-join evaluation in massively parallel systems, Sigmod Record 2016 (not yet published).