Parallel-Correctness and Transferability for Conjunctive Queries

Tom J. Ameloot$^1$ Gaetano Geck$^2$ Bas Ketsman$^1$
Frank Neven$^1$ Thomas Schwentick$^2$

$^1$ Hasselt University $^2$ Dortmund University
Big Data

“Too large for one server”

Several systems: Hadoop, Spark, . . . many others

Common Strategy

- Data is distributed
- Query evaluation: Multiple rounds with reshuffling
Simple Evaluation Algorithm

1-Round MPC model \[\text{Koutris & Suciu 2011}\]

Input = query $Q$

Modeled by a distribution policy $P$

Step 1:

Redistribution

Step 2:

Output = union of output at each server
Main Problems

Semantical correctness:

When is the simple algorithm correct on a distribution policy?

Parallel-Correctness

Multiple-query optimization:

Which queries allow to reuse the distribution obtained for another query?

Transferability

Formal framework for reasoning about correctness of query evaluation and optimization in a distributed setting
Outline

1. Definitions
2. Parallel-Correctness
3. Transferability
4. Lowering the Complexity
5. Conclusion & Future Work
Definitions

Database schema

Infinite set of data values

Instance $I$ is a finite set of facts $R(d_1, \ldots, d_n)$

Conjunctive Query: $T(\bar{x}) \leftarrow R_1(\bar{y}_1), \ldots, R_m(\bar{y}_m)$
Network $\mathcal{N}$ is a finite set of nodes.

**Definition**

A distribution policy $P$ is a total function mapping facts (over $\text{dom}$) to sets of nodes in $\mathcal{N}$.
Distribution Policies

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

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

- Invented in the context of Datalog evaluation
  [Ganguli, Silberschatz & Tsur 1990]
- Described in Map-Reduce context
  [Afrati & Ullman 2010]
- Intensively studied
  [Beame, Koutris & Suciu 2014]

Algorithm:

- Reshuffling based on structure of $Q$
  Partitioning of complete valuations over servers in instance independent way through hashing of domain values
Simple Evaluation Algorithm

Input = query $Q$

Step 1: distribute data over servers w.r.t. $P$

Step 2: evaluate $Q$ at each server
Parallel-Correctness

**Definition**

\( Q \) is parallel-correct on \( I \) w.r.t. \( P \), iff

\[
Q(I) = \bigcup_{\kappa \in \mathbb{N}} Q(\text{dist}_{P,I}(\kappa))
\]

\( \supseteq \) by monotonicity

**Definition (w.r.t. all instances)**

\( Q \) is parallel-correct w.r.t. \( P \) iff

\( Q \) is parallel-correct w.r.t. \( P \) on every \( I \)
Parallel-Correctness
Sufficient Condition

(C0) for every valuation $V$ for $Q$,

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

**Intuition:** Facts required by a valuation meet at some node

**Lemma**

(C0) implies $Q$ parallel-correct w.r.t. $P$.

Not necessary
(C0) not Necessary

Example

Distribution policy $P$

$$all \setminus \{R(a, b)\}$$

$$all \setminus \{R(b, a)\}$$

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:

$R(a, a)$

Derives:

$T(a, a)$

$\supseteq$
Parallel-Correctness
Characterization

**Lemma**

\( Q \) is parallel-correct w.r.t. \( P \) iff

\[(C1) \quad \text{for every minimal valuation } V \text{ for } Q,\]

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

**Definition**

\( V \) is **minimal** if no \( V' \) exists, where

\[V'(\text{head}_Q) = V(\text{head}_Q), \quad V'(\text{body}_Q) \subsetneq V(\text{body}_Q).\]
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}

\supseteq

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

\text{Derives:}

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

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

Requires:

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

\text{Derives:}

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

\text{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 $\Pi^P_2$-complete.

Proof:
- Lower bound: Reduction from $\Pi_2$-QBF
- Upper bound: Characterization
  but, requires proper formalization of $P$
Outline

1. Definitions
2. Parallel-Correctness
3. Transferability
4. Lowering the Complexity
5. Conclusion & Future Work
Computing Multiple Queries

\[ Q \rightarrow \]

\[ Q(I) \leftarrow \]

\[ Q' \rightarrow \]

\[ Q'(I) \leftarrow \]

\[ Q \rightarrow \]

\[ Q(I) \leftarrow \]

\[ Q' \rightarrow \]

\[ Q'(I) \leftarrow \]
When can $Q'$ be evaluated on distribution used for $Q$?

$Q' \rightarrow$  

No reshuffling

$Q'(I) \leftarrow$
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.

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

Based on query structure alone, not on distribution policies
Lemma

\( Q \xrightarrow{T} Q' \) 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 \( Q \xrightarrow{T} Q' \) is \( \Pi_3^P \)-complete.

- Lower bound: Reduction from \( \Pi_3\text{-QBF} \)
- Upper bound: Characterization
Outline

1. Definitions

2. Parallel-Correctness

3. Transferability

4. Lowering the Complexity

5. Conclusion & Future Work
Strongly Minimal CQs

Definition
A CQ is strongly minimal if all its valuations are minimal.

- Full-CQs
  \[ T(x, y) \leftarrow R(x, y), R(x, x) \]

- CQs without self-joins
  \[ T() \leftarrow R(x, y), S(x, x) \]

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

A minimal CQ is not always strongly minimal.
Strongly Minimal CQs

Lemma
Deciding whether $Q$ is strongly minimal is coNP-complete

Theorem
Deciding $Q \rightarrow_T Q'$ is NP-complete for strongly minimal $Q$
Hypercube

Algorithm:

- Reshuffling based on structure of $Q$

  Partitioning of complete valuations over servers in instance independent way through hashing of domain values

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

Definition

$Q \xrightarrow{H} Q'$ 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 node ($\forall P \in \mathcal{H}(Q)$)

▸ **$Q$-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)
Related Concepts

Containment

\[ Q \subseteq Q' \]

Lemma

Containment and transferability are incomparable

Determinacy (Data-Integration)

\[ Q'(I) = Q'(J) \text{ implies } Q(I) = Q(J), \text{ for every } I, J \]

Lemma

Determinancy and transferability are incomparable
Summary

Formal framework for reasoning about correctness of query evaluation and optimization in a distributed setting

Main concepts:
- Parallel-correctness
- Transferability

Independent of expression mechanism
Future Work

Expression Formalism for distribution policies

▸ Other than Hypercube?

Distribution policy for set of queries

▸ Given CQ: which distribution policy? Hypercube

▸ Given set of CQs: which distribution policy? Open question
Future Work

Tractable Results

► Other classes of queries?
► Other families of distribution policies?

More expressive classes of queries

► This work: CQs
► FO: undecidable
► initial results: UCQs, CQs with negation