A Worst-Case Optimal Multi-Round Algorithm for Parallel Computation of Conjunctive Queries

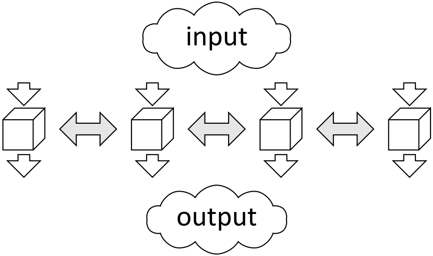
Bas Ketsman & Dan Suciu

Topic of the Talk

How to compute **multi-joins** (over graphs) ...

$$(x, y, z) \leftarrow R(x, y), S(y, z), T(z, x)$$

... in a multi-round **shared nothing cluster** setting ...



... with **communication cost** that is **worst-case optimal**?

Introduction

Worst-case optimality:

- **Output size:** AGM bound [Atserias, Grohe & Marx 08]

  query output $= m^{\rho^*}$.

  $\boxed{\text{Lower-bound on worst-case running-time}}$

- **Optimal sequential algorithms:** (w.r.t running-time)

  Leapfrog-trie-join, NPRR, Generic Join

Introduction (2)

Worst-case optimal communication cost:

- **Load** = maximal amount of messages received by any server in any communication round

- **Lowerbound**

  load $\geq \frac{m}{p^{1/\rho*}}$. [Koutris, Beame & Suciu 16]

- **Optimal parallel algorithms:** (w.r.t communication cost) [Koutris, Beame & Suciu 16]

  Ad-hoc algorithms for chains, stars, simple cycles

Main Result

A parallel algorithm exists for computing join queries over graphs using only a constant number of rounds and

$$\text{load} \leq \tilde{\mathcal{O}}(m/p^{1/\rho^*}).$$

Query/schema restrictions:

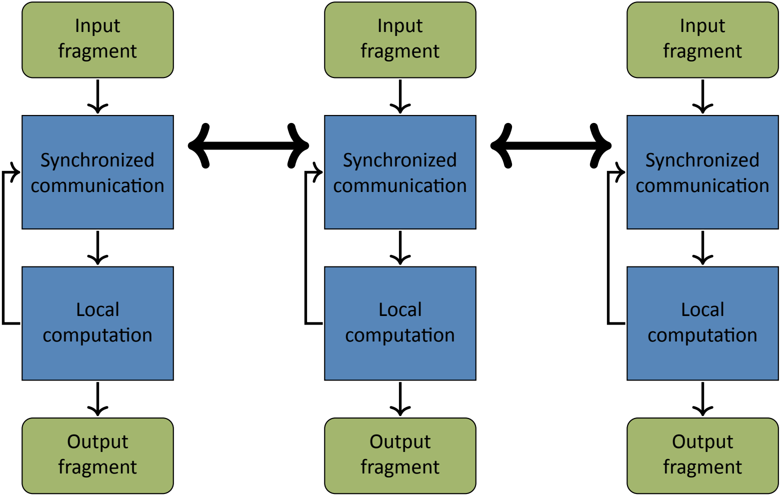
- Arity at most two
- No projections
- No self-joins

Essentially optimal:

- Up to a poly-log factor
- Data-complexity

Outline

Massively Parallel Communication Model: [Koutris, Suciu 2011]



Outline

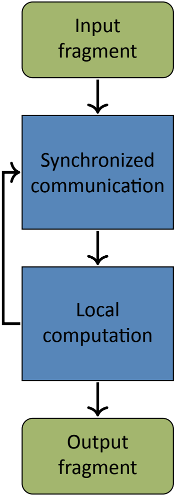
The Model

Lowerbound and Hypercube (ρ^* and τ^*)

Main Result by Example

Summary & Future Work

Lower-Bound [Koutris, Beame, Suciu 2016]



For a constant-round algorithm to be correct for given query on every instance

worst-case load is

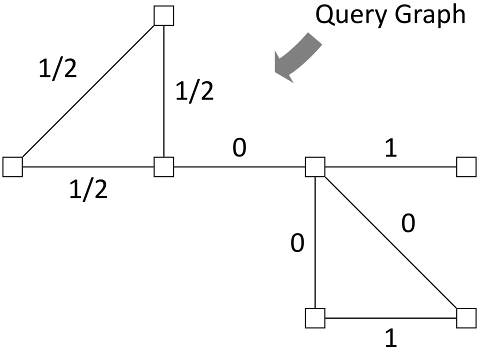
$$\geq \frac{m}{p^{1/\rho^*}}$$

(assuming equi-sized relations)

Through AGM bound

ρ^* = Fractional Edge Covering Number

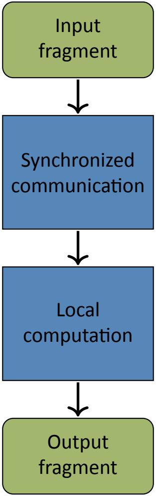
$R_1(x, y), R_2(y, z), R_3(z, x), R_4(z, u), R_5(u, w), T_6(u, t), T_7(t, s), T_8(s, u)$



$\rho^* = 7/2$

- **Objective function:** Assign a positive weight to every edge
- **Constraint:** Every vertex incident to sum of weights ≥ 1
- **Optimization goal:** Minimize total sum of assigned weights

Hypercube (= shares algorithm)



= Single-round hash-join algorithm

Introduced by [Afrati, Ullman, 2010]

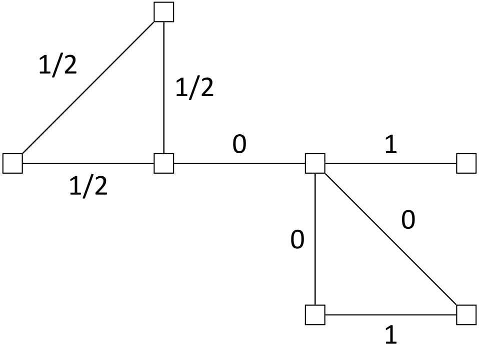
If **database has no skew**, runs with load:

$$\leq \frac{m}{p^{1/\tau^*}}$$

(w.h.p. and ignoring poly-log factor)

[Beame, Koutris, Suciu 2013]

τ^* = Fractional Edge **Packing** Number



$\tau^* = 7/2$

- **Objective function:** Assign a positive weight to every edge
- **Constraint:** Every vertex incident to sum of weights ≤ 1
- **Optimization goal:** **Maximize** total sum of assigned weights

Relation between τ^* and ρ^*?

Solution is tight if satisfies $=$ rather than $\leq$ or $\geq$.

For general hypergraphs: No clear relation between τ^* and ρ^*!

For simple graphs:

- Optimal half-integral fractional edge packings exist (using only weights 1, $1/2$ and 0)
- $\tau^* \leq \frac{|\text{vars}(\mathcal{Q})|}{2} \leq \rho^*$ (assign weights 1/2 to all vertices)
- $\tau^* + \rho^* = |\text{vars}(\mathcal{Q})|$

Outline

The Model

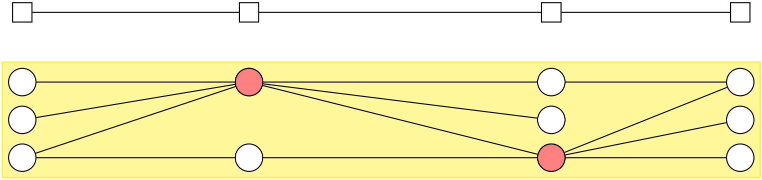
Lowerbound and Hypercube (ρ^* and τ^*)

Main Result by Example

Summary & Future Work

Heavy-Hitter Configurations

Example Query:
$(x, y, z) \leftarrow R_1(x, y), R_2(y, z), R_3(z, u)$



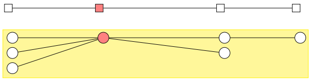
Heavy-hitter: value with degree $> \delta$ (in some direction)

Skew: some heavy-hitter exists

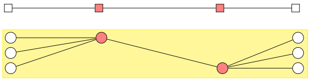
Break Skewed Instance in Understandable Pieces

Heavy-hitter configuration (δ, H): A skew threshold value δ + labeling of query variables with "heavy" (H) or "light" (others).

1.



2.



Matching instance $I_{(\delta,H)} =$ induced subinstance where heavy variables have only the heavy values, light variables only the light values.

Break Skewed Instance in Understandable Pieces

Evaluation strategy:
Compute Q in parallel over all instances $I_{(\delta,H)}$ using the same p servers.

For Fixed δ:

Claim: $\bigcup_{H \subseteq vars(Q)} Q(I|_{(\delta,H)}) = Q(I)$.

As the number of configurations depends on Q,
maximal load $\leq \max_H\{$**maximal load** to compute Q on $I_{(\delta,H)}\}$.

(ignoring constants)

The Algorithm in a Nutshell

Preprocessing:

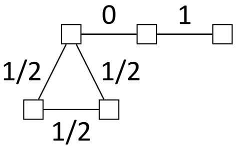
- *Identify where skew is*
  Heavy-hitters and degrees of heavy-hitters.

Algorithm:

1. Break skewed instance in understandable pieces
2. Divide and Conquer strategy to deal with skew
3. Solve remaining (skew-free) problem with Hypercube

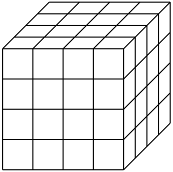
The Algorithm by Example

Example query



- $\tau^* = \rho^* = |\mathsf{vars}(\mathcal{Q})|/2$

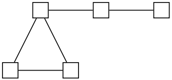
Servers



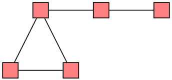
The Algorithm by Example

Threshold value: $\delta = \frac{m}{p^{1/|\mathsf{vars}(\mathcal{Q})|}}$

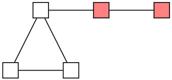
Do computation for each heavy-hitter configuration in parallel



"all light"

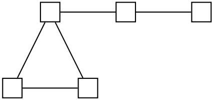


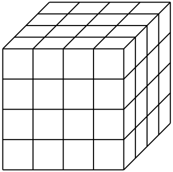
"all heavy"



"hybrid"

The Algorithm by Example: “All light”

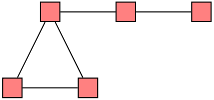


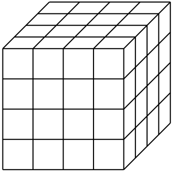


Use the Hypercube algorithm

- Due to tightness: $\tau^* = \rho^* = |\mathsf{vars}(\mathcal{Q})|/2$
- non skewed means: degree $\leq \delta = \frac{m}{p^{1/|\mathsf{vars}(\mathcal{Q})|}} = \frac{m}{p^{1/(2\tau^*)}}$
- Hypercube ensures load $\leq \frac{m}{p^{1/\tau^*}} = \frac{m}{p^{1/\rho^*}}$.

The Algorithm by Example: “All heavy”



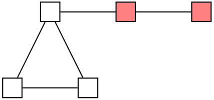


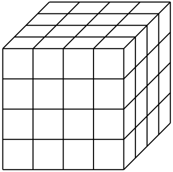
Broadcast all relations

- A value is heavy if degree $> \delta = \frac{m}{p^{1/|\mathsf{vars}(\mathcal{Q})|}}$.
- An heavy attribute has $\leq p^{1/|\mathsf{vars}(\mathcal{Q})|}$ heavy values.
- A heavy relation has $\leq p^{2/|\mathsf{vars}(\mathcal{Q})|}$ heavy tuples.
- Every server receives at most $p^{2/|\mathsf{vars}(\mathcal{Q})|}$ tuples.
- $p^{2/|\mathsf{vars}(\mathcal{Q})|} \leq \frac{m}{p^{2/|\mathsf{vars}(\mathcal{Q})|}} = \frac{m}{p^{1/\rho^*}}$ due to $m \geq p^2$.

(ignoring the constants)

The Algorithm by Example: "Hybrid"

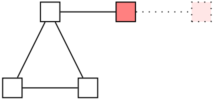


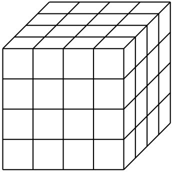


Step 1: Broadcast heavy relation

- As before: load $\leq \frac{m}{p^{1/\rho^*}}$ due to $m \geq p^2$.

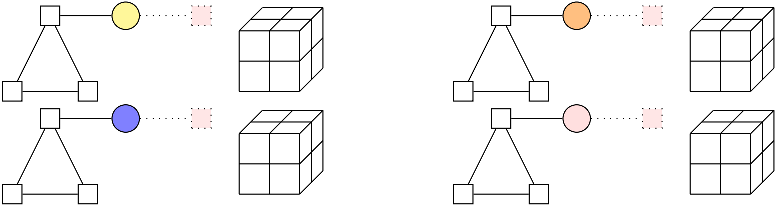
Refocus:



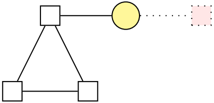


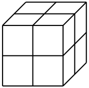
- Solution can be easily extended.

Step 2: Assign group of servers to every heavy value



- Combination of outputs = complete output



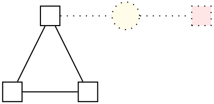


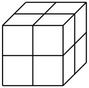
- size of group $p' = p^{(|\mathsf{vars}(\mathcal{Q})|-1)/|\mathsf{vars}(\mathcal{Q})|}$
  (because $\leq p^{1/|\mathsf{vars}(\mathcal{Q})|}$ heavy values)

Step 3: Semi-join reduce involved relations

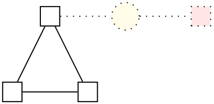
- reductions are cheap: 2 rounds and load $\leq \frac{m}{p'} \leq \frac{m}{p^{1/\rho^*}}$
  (because we have > 2 light variables)

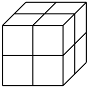
Refocus:





- Output for simpler query can be translated to output for original query by simply adding to every tuple the locally known heavy value





Step 4: Hypercube

- degrees $\leq \frac{m}{p^{1/|\mathsf{vars}(\mathcal{Q})|}} = \frac{m}{p'^{1/(|\mathsf{vars}(\mathcal{Q})|-1)}} \leq \frac{m}{p'^{1/|\mathsf{vars}(\mathcal{Q}')|}} = \frac{m}{p'^{1/(2\tau^*(\mathcal{Q}'))}}$
- Hypercube guarantees load $\leq \frac{m}{p'^{1/\tau^*(\mathcal{Q}')}} \leq \frac{m}{p^{1/\rho^*(\mathcal{Q})}}$

$\boxed{\text{done}}$

Sometimes more complex: algorithm uses up to 9 rounds

Outline

The Model

Lowerbound and Hypercube (ρ^* and τ^*)

Main Result by Example

Summary & Future Work

Main Result

Every conjunctive query without self-joins, that is full, over relations with arities at most two can be computed in 9 rounds with load $\leq \tilde{\mathcal{O}}(\frac{m}{p^{1/\rho^*}})$.

Essentialy optimal

ρ^* seems the right way to express optimality for the communication cost of distributed query evaluation algorithms, at least when relation arities do not exceed two.

Future Work

Does an algorithm exist with worst-case optimal load $m/p^{1/\rho^*}$ for queries over relations with arbitrary-arities?

- relation between edge cover / packing unclear in general
- half-integral edge cover/packing does not always exist
- queries exist where $\tau^* > \rho^*$

  $R_1(x_1, y_1, z_1), R_2(x_2, y_2, z_2), S_1(x_1, x_2), S_2(y_1, y_2), S_3(z_1, z_2).$

  $\Rightarrow$ Hypercube cannot be used even when there is no skew

Is $m/p^{1/\rho^*}$ a tight lowerbound for joins over arbitrary-arity relations?

Future Work (2)

Are the 9 rounds essential?

What if queries have existential quantification (projections)?

What if the database has dependencies?

Thank you!