Cross-Model Conjunctive Queries over Relation and Tree-structured Data

DASFAA 2022

Yuxing Chen, Valter Uotila, Jiaheng Lu
University of Helsinki
Zhen Hua Liu, Souripriya Das
Oracle

CONTENTS

01 Motivation

02 Contribution

03 Application

Motivation

- Big data (Variety) => connect information
- Different data formats (Relational, tree data) => cross-model join
- Worst-case join optimal join (less intermediate result in worst case)

Motivation

- Theoretical interest
  - Q1: Size bound for tree pattern
  - Q2: New worst-case optimal algorithms
- Application
  - Data lake
  - Polystores or multi-model databases

Size bound of query

- In AGM worst-case model, a table's size is the full combination of the attributes' unique values.
- Find the size bound of R(A,B,C) (O(N^2))
- by R(A,B) ⋈ R(B,C) (O(N) for each table size)

Find size bound: Max $N_A * N_B * N_C$

Bound tables:

$$N_A * N_B \leq N$$

$$N_B * N_C \leq N$$

Solution:

$$N_A * N_B * N_C = N^2$$

Unique values:

By: $N_A = N_C = N$

$N_B = 1$

$O(N)$

A	B
a_1	b_1
…	…
a_n	b_1

⋈

$O(N)$

B	C
b_1	c_1
…	…
b_1	c_n

Linear programming simplicity

- For simplicity, do logarithm ($\log_N$) for both sides.

Max $N_A * N_B * N_C$	Max $A + B + C$
$N_A * N_B \leq N$	$A + B \leq 1$
$N_B * N_C \leq N$	$B + C \leq 1$

$N_A * N_B * N_C = N^2$

By:

$N_A = N_C = N$

$N_B = 1$

$A + B + C = 2$

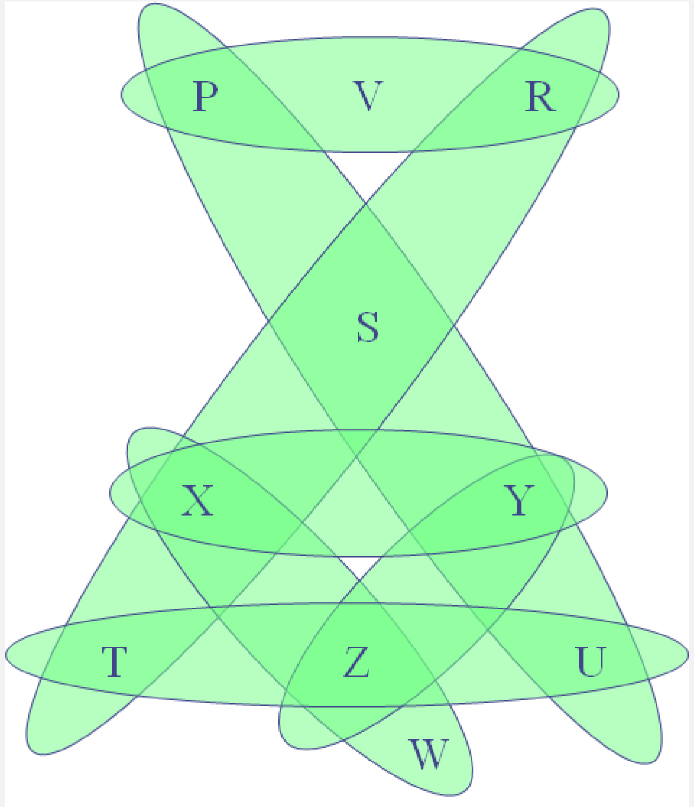
By:

$A = C = 1$ — N^1 Unique values

$B = 0$ — N^0 Unique values

Complex query example

- By AGM, we can
  - Find size bound
  - Construct worst-case instance
  - $O(N^4)$



Max $P + V + R + S + X + Y + T + Z + U + W$

$$P + V + R \leq 1$$
$$P + S + Y + U \leq 1$$
$$R + S + X + T \leq 1$$
$$X + Y \leq 1$$
$$X + Z + W \leq 1$$
$$Y + Z \leq 1$$
$$T + Z + U \leq 1$$

Tree pattern query (tree join)

- Tree data: XML, JSON
- Pattern query:
  - “A/B”, find value matching where A is the parent of B
  - “A//B”, find value matching where A is the ancestor of B

PC path

AD path

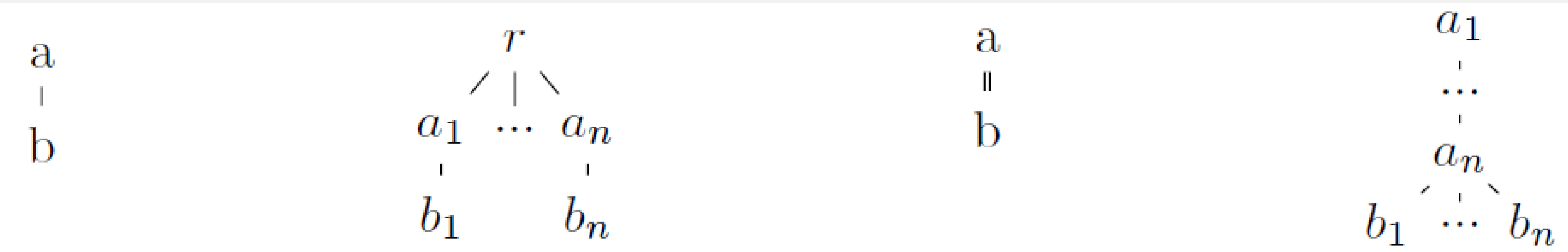
```
{
‘A’:”a1”
‘C’:”c1”
‘A’:{
      ‘B’:”b1”
      }
}
```

JSON1

```
{
‘A’:{
      ‘C’:{‘B’:”b1”}
      }
}
```

JSON2

Worst case for tree pattern



(a) Child axe (b) Tree instance (c) Descendant axe (d) Tree instance

- PC path: $O(N)$ => constraint $a + \mathrm{b} \leq 1$
- AD path: $O(N^2)$ => constraint $a \leq 1$; $\mathrm{b} \leq 1$
- We assume the tree instance or node of each attribute is O(N).

Size bound for tree pattern

$a + b \leq 1$

$a + c \leq 1$

$d \leq 1$

$O(N^3)$

Not correct

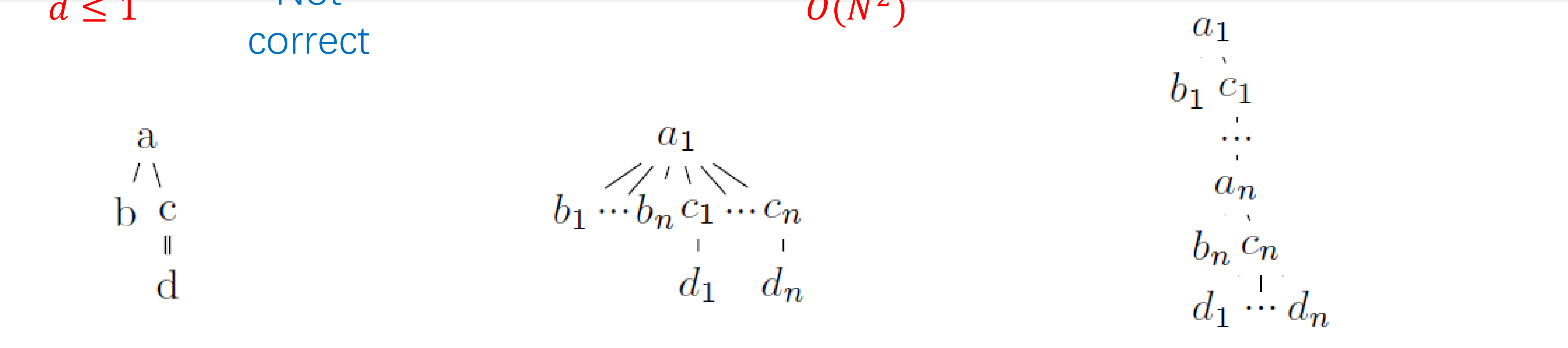
$a + b \leq 1$

$a + c + d \leq 1$

$O(N^2)$

$a + b + c \leq 1$

$d \leq 1$



(a) Tree query (b) Tree instance 1 (c) Tree instance 2

- However when combining branches and AD:
  - More constraints than PC and AD
  - Alternative constraints

Find size bound for tree pattern

- Size bound
  - PC is constrained to O(N)
  - AD is not constrained
  - Alternatives when with branches and AD path
- The size bound is the maximum among all alternatives

Worst case optimal join

- One attribute at a time
- Compute result by all related tables
- R(A,B) ⋈ R(B,C) ⋈ R(A,C)
- Property: projection result is always no more than final result with the same constraints

Max $A + B + C$

$A + B \leq 1$

$B + C \leq 1$

$A + C \leq 1$

Max $\mathrm{A\ or\ B\ or\ C} \leq 1 \leq 3/2$
Max $\mathrm{A} + \mathrm{B} \leq 1 \leq 3/2$
Max $\mathrm{B} + \mathrm{C} \leq 1 \leq 3/2$
Max $\mathrm{A} + \mathrm{C} \leq 1 \leq 3/2$
Max $\mathrm{A} + \mathrm{B} + \mathrm{C} \leq 3/2$

Worst case optimal join

- R1(A,B) ⋈ R2(B,C) ⋈ R3(A,C) solution1:
  - Step1: Find result of R(A) by R1,R3
  - Step2: Use R(A) to find result of R(A,B) by R1,R2
  - Step3: Use R(A,B) to find result of R(A,B,C) by R2,R3

Max $A + B + C$

$A + B \leq 1$

$B + C \leq 1$

$A + C \leq 1$

Max $\mathrm{A} \leq 1 \leq 3/2$
Max $\mathrm{A} + \mathrm{B} \leq 1 \leq 3/2$
Max $\mathrm{A} + \mathrm{B} + \mathrm{C} \leq 3/2$

Worst case optimal join for CMCQs

- Challenges
  - Decode and use positions to match
  - Positions are irrelevant to final result
  - Some patterns (multiple branches, ancestor-descendant matching) are easy to scale massive result
- Contribution:
  - Path result first and remove the irrelevant position values
  - Join simultaneously both relations and PC-path

CMJoin Algorithm

- Tree data => intermediate data
- Path result => project-out non-branch position value
- Generic worst-case optimal join between tree data and relational data

---

Algorithm 1: *CMJoin*

Input: Relational tables $\mathcal{R}$, pattern queries $\mathcal{T}$

$\mathcal{R}' \leftarrow \emptyset$ `// Tree intermediate result`

if $\rho_1(S_r \cup S_p) \leq \rho_3(S_r)$ **then** `// Theorem 1 condition (1)`

  foreach $N \in \mathcal{T}$ **do**

    $\mathcal{R}' \leftarrow \mathcal{R}' \cup C_N(r_N, p_N)$ `// Nodes as tables`

else

  $\mathcal{P} \leftarrow \mathcal{T}.getPaths()$

  foreach $P \in \mathcal{P}$ **do**

    $R_P(S_r \cup S_p) \leftarrow$ path result of P `// Paths as tables`

    $R'_P(S_r \cup S'_p) \leftarrow$ project out non-branch position values of $R_P(S_r \cup S_p)$

    $\mathcal{R}' \leftarrow \mathcal{R}' \cup \{R'_P(S_r \cup S'_p)\}$

$Q(S_r \cup S'_p) \leftarrow generic\_join(\mathcal{R} \cup \mathcal{R}')$

$Q(S_r) \leftarrow$ project out all position values $Q(S_r \cup S'_p)$

Output: Join results $Q(S_r)$

Evaluation

- Diverse datasets and complex queries
- Less intermediate result => faster respond time

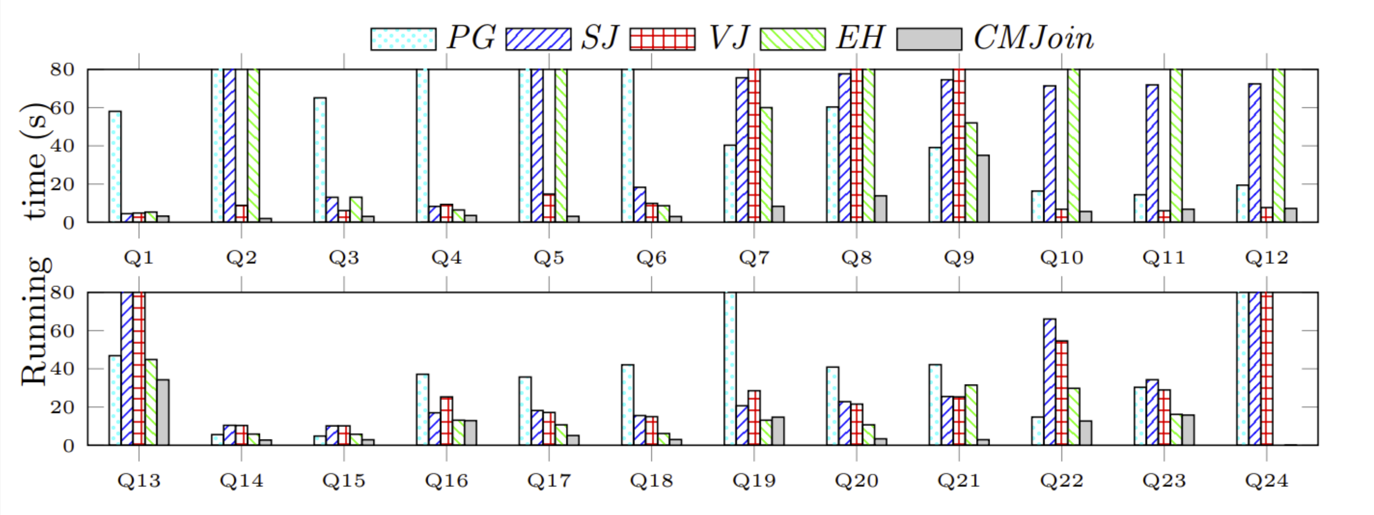


Table 5.2: Intermediate result size (10^6) and running time (S) for queries. "/" and "-" indicate "time out" ($\geq$ 10 mins) and "out of memory", respectively.

	Intermediate result size(10^6)					Running time (S)					
Query	**PG**	**SJ**	**VJ**	**EH**	***CMJoin***	**CDB**	**PG**	**SJ**	**VJ**	**EH**	***CMJoin***
Q1	7.87x	2.60x	2.00x	1.68x	0.15	153.5x	18.02x	1.39x	1.51x	1.66x	3.22
Q2	/	-	3.75x	4.83x	0.08	/	/	-	4.52x	129x	1.96
Q3	86.0x	62.6x	3.63x	4.61x	0.08	/	21.3x	4.27x	1.99x	4.28x	3.06
Q4	/	1.96x	1.75x	1.64x	0.24	/	/	2.34x	2.63x	1.82x	3.55
Q5	/	-	1.86x	1.77x	0.22	/	/	-	4.75x	39.8x	3.11
Q6	/	2.24x	2.00x	1.85x	0.21	/	/	6.10x	3.30x	2.89x	3.00
Q7	133x	106x	-	35.1x	0.29	15.4x	4.82x	9.05x	-	7.18x	8.36
Q8	350x	279.8x	-	/	0.11	/	4.36x	5.61x	-	/	13.8
Q9	8.87x	8.34x	-	2.01x	4.62	2.20x	1.12x	2.13x	-	1.48x	35.0
Q10	110x	440x	4.86x	/	0.07	/	2.91x	12.7x	1.22x	/	5.62
Q11	110x	440x	4.86x	/	0.07	/	2.11x	10.5x	0.88x	/	6.84
Q12	110x	440x	4.86x	/	0.07	/	2.68x	9.99x	1.06x	/	7.25
Q13	1.04x	1.22x	1.22x	1.07x	43.2	6.87x	1.37x	4.81x	4.79x	1.31x	34.2
Q14	19.7x	2.56x	2.56x	3.90x	0.39	3.22x	2.04x	3.82x	3.79x	2.14x	2.73
Q15	14.2x	1.85x	1.85x	17.0x	0.54	1.40x	1.68x	3.53x	3.54x	2.01x	2.87
Q16	1.24x	1.24x	6.81x	2.15x	0.37	/	2.88x	1.32x	1.96x	1.02x	12.9
Q17	1.59x	7.84x	2.28x	1.31x	0.32	/	7.03x	3.58x	3.38x	2.10x	5.08
Q18	1.59x	7.13x	1.59x	1.64x	0.32	/	14.1x	5.21x	5.02x	2.06x	2.98
Q19	/	5.47x	6.62x	1.77x	0.45	/	/	1.41x	1.94x	0.89x	14.7
Q20	7.80x	25.1x	7.30x	4.19x	0.10	/	12.1x	6.77x	6.39x	3.17x	3.37
Q21	12.0x	36.1x	18.4x	14.7x	0.10	/	14.6x	8.82x	8.75x	10.9x	2.89
Q22	1.00x	18.5x	18.5x	0.96x	0.57	1.47x	1.16x	5.22x	4.31x	2.35x	12.7
Q23	18.5x	18.5x	18.5x	1.61x	0.57	1.46x	1.92x	2.17x	1.83x	1.02x	15.8
Q24	14.3x	3.02x	4.02x	0.96x	0.57	>4kx	>9kx	>11kx	>12kx	0.18x	0.01
AVG	5.46x	5.90x	1.92x	1.90x	2.24	13.43x	4.37x	5.34x	3.33x	3.46x	8.54

Application

- Mostly theoretical interests
- Yet
  - Like nested loop can be speeded up by Hash join
  - Complex queries can be processed in worst-case optional way to avoid huge intermediate results
- Current industry
  - MarkLogic

Thank you

More questions:

yuxing.chen@helsinki.fi

Reference

- **AGM boud:** Atserias, Albert, Martin Grohe, and Dániel Marx. "Size bounds and query plans for relational joins." 2008 49th Annual IEEE Symposium on Foundations of Computer Science. IEEE, 2008.
- **Worst-case optimal join:** Ngo, Hung Q., Ely Porat, Christopher Ré, and Atri Rudra. "Worst-case optimal join algorithms." Journal of the ACM (JACM) 65, no. 3 (2018): 1-40.