Introduction to KNU DBLab and Robust Tuple Extraction

July 10, 2009

Wook-Shin Han

Recent Publications

- Query Optimization
  - Progressive Optimization [SIGMOD'07, VLDB'06]
  - Parallelizing Query Optimization [SIGMOD'09, VLDB'08]

- Query Processing
  - Similarity Search in *Large* Time-Series Data [VLDB'07, SIGMOD'01]
  - Applying DB Query Processing to *Massive* Amount of Streaming Data
    - Streaming XML Processing [VLDB'08, WWW'07]
    - Streaming Moving Object Processing [IEEE TKDE'09]

Results for

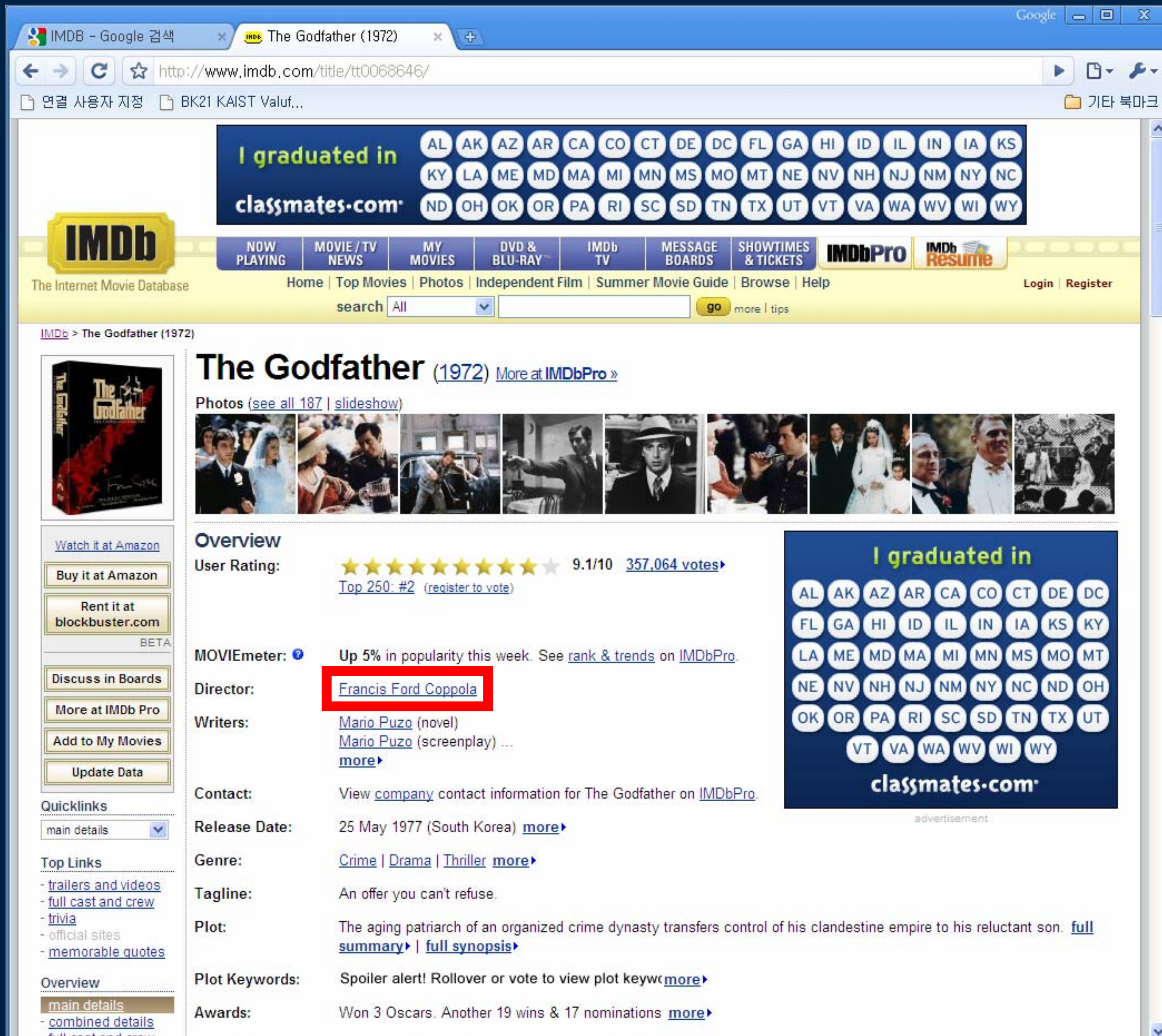
- 1 conference paper published (SIGMOD 2009)
- 1 conference paper submitted

Tuple Extraction from Web Pages

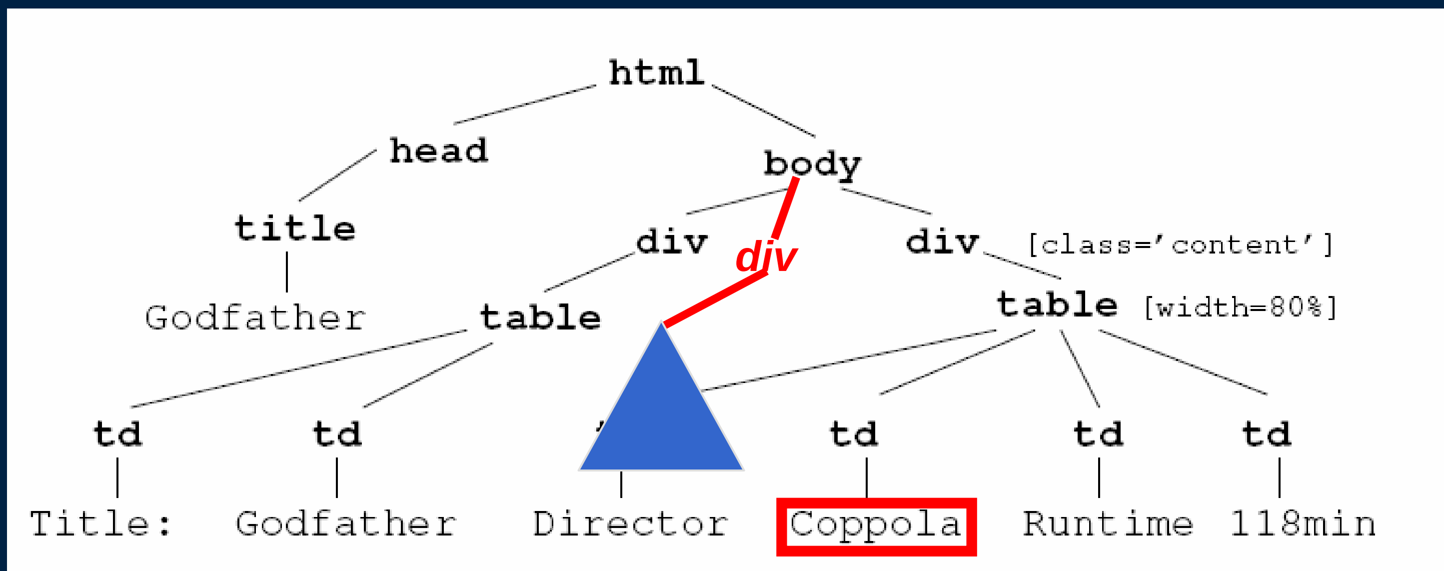
- Extracting tuples from HTML pages has been an important issue in various applications

- Applications
  - Web data integration
  - E-commerce market monitoring
  - Mashups

An Example Web Page



A Web Page at Time T_0



$$W_1 \equiv \texttt{/html/body/div[2]/table/td[2]/text()}$$

http://portal.acm.org/ft_gateway.cfm?id=1559882&type=pdf&coll=GUIDE&dl=&CFID=45905101&CFTOKEN=56901971

Motivation

- The current state-of-the-art extraction systems are *very vulnerable* to small changes on the web

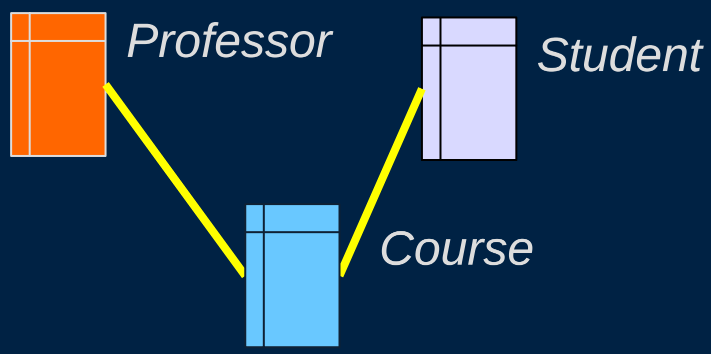
- A robust extraction solution is needed!

Dependency-Aware Reordering for Parallelizing Query Optimization (SIGMOD 2009)

An example query

- SQL:

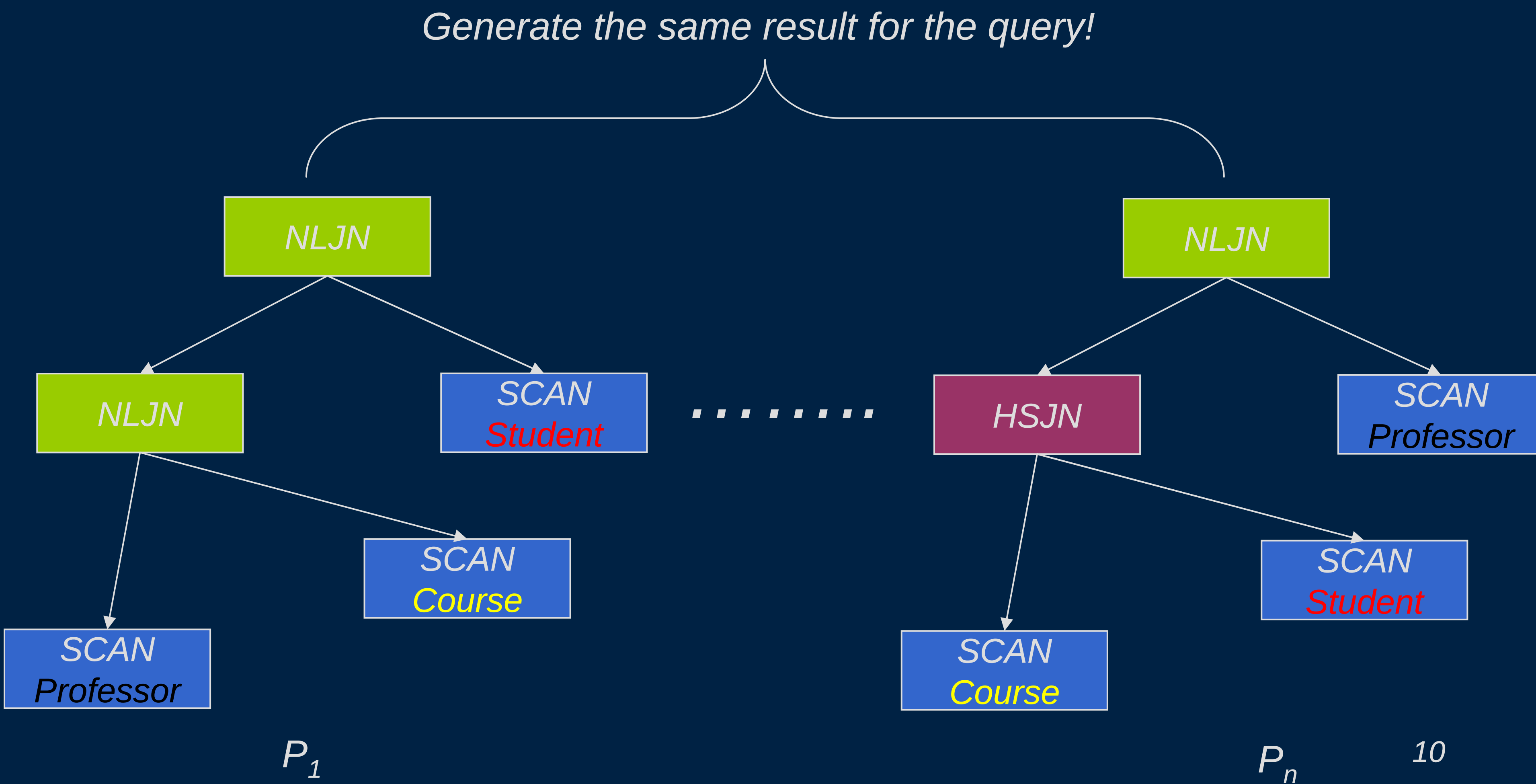
```
    SELECT *
      FROM Professor P, Course C, Student S
     WHERE P.pid = C.pid
       AND S.sid = C.sid
```



Only need to specify ***what*** *to find!*

Query Execution Plans (QEPs)

Plan = Query Execution Plan (QEP) = Access Plan



Example: Query Optimization

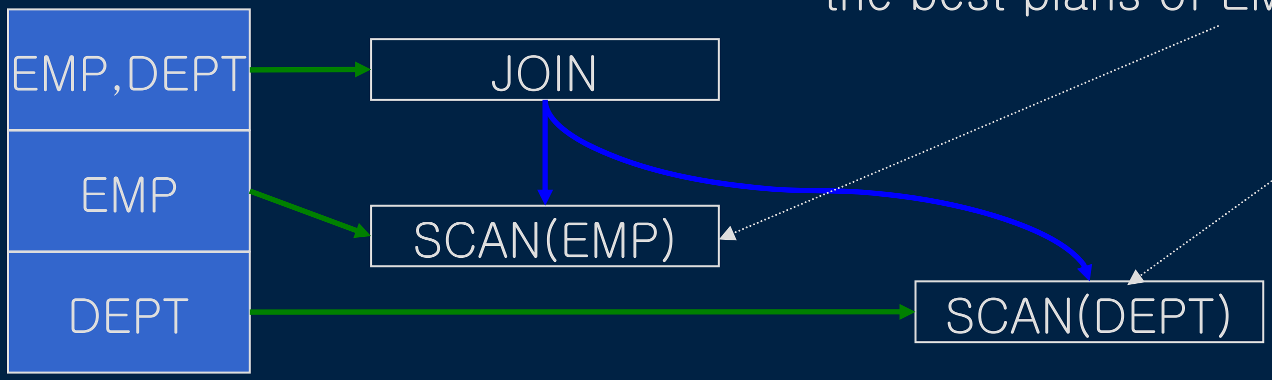
SELECT name, mgr
FROM EMP e, DEPT d
WHERE e.dno=d.dno and
e.sal > 300000

quantifier

Level 1:
Generate best (table access) plan for EMP
Generate best (table access) plan for DEPT

Level 2:
Generate best (join) plan using
the best plans of EMP and DEPT

MEMO table (HASH table)

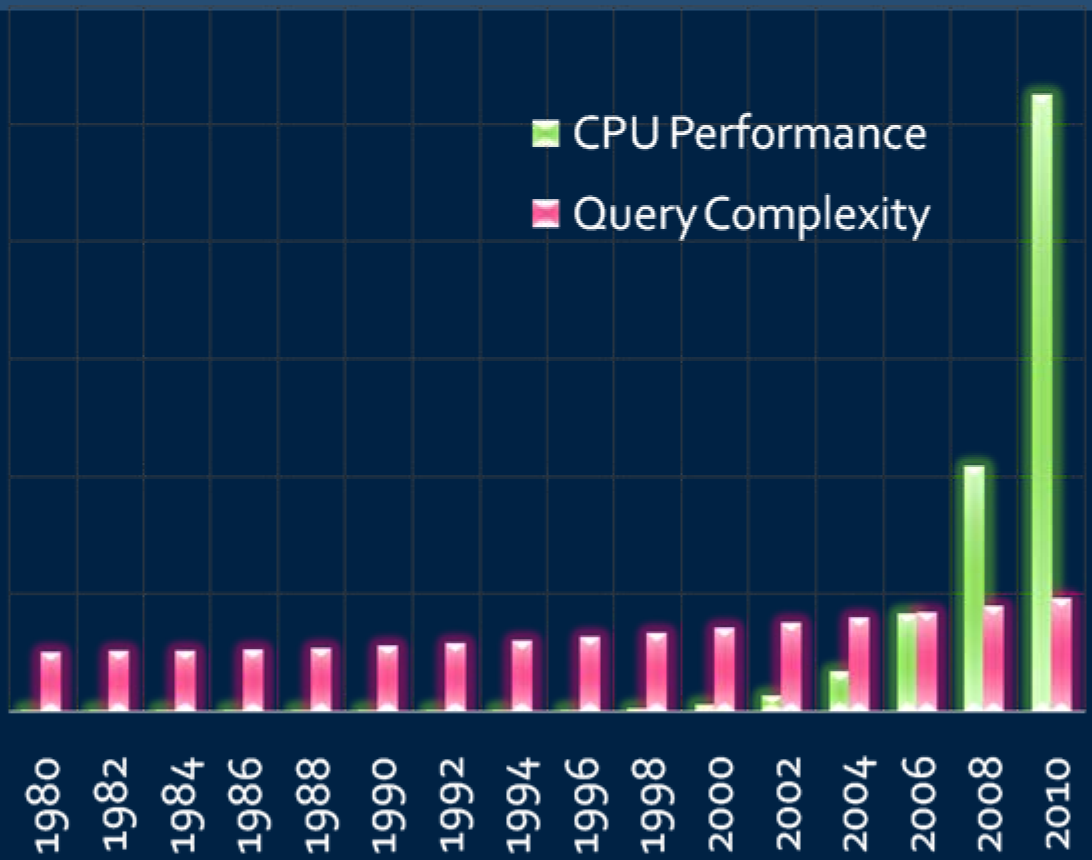


DP Query Optimization

- Query optimization exploits dynamic programming (DP) to avoid generating redundant QEPs.

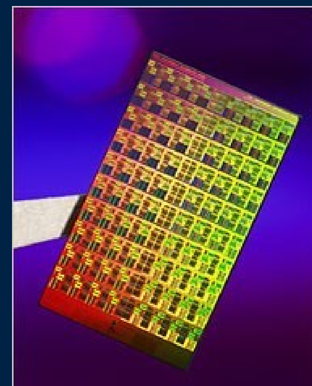
- Query optimization times using DP can increase significantly as the number of joins in a query increases.

Query Complexity vs. CPU Performance



Moore's Law outperforms query complexity

Multi-Core Wave



intel Core 2 Duo inside

intel Quad Core

intel 8-Core

intel Terascale

AMD 64 Opteron

AMD Quad Core

AMD 8-Core

Supporting up to 256 cores

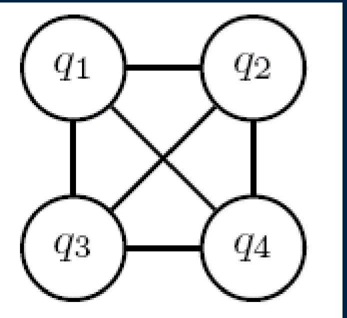
Motivation

- Parallelizing query optimization significantly delays the need to rely on suboptimal heuristics

- Our previous work on parallelizing query optimization [VLDB08] has some limitations
  - *No support* for recently developed enumeration algorithms
  - *Static search space allocation* can lead to very unbalanced workloads
  - Does *not fully exploit parallelism* due to local memo merge

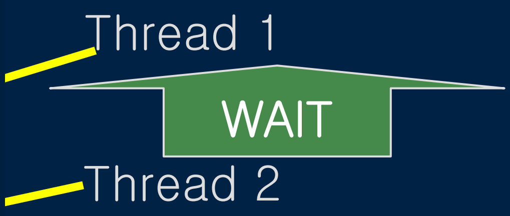
Example of DPccp Optimizer

SELECT *
FROM R_1 q_1, R_2 q_2,
R_3 q_3, R_4 q_4
WHERE $q_1.a_2 = q_2.a_1$ and
$q_1.a_3 = q_3.a_1$ and
$q_1.a_4 = q_4.a_1$ and
$q_2.a_3 = q_3.a_2$ and
$q_2.a_4 = q_4.a_2$ and
$q_3.a_4 = q_4.a_3$



Query graph G

1. (q_4)
2. (q_3)
3. (q_2)
4. (q_1)
5. (q_3, q_4)
6. (q_2, q_4)
7. (q_2, q_3)
8. (q_2, q_3q_4)
9. (q_4, q_2q_3)
10. (q_3, q_2q_4)
11. (q_1, q_4)
12. (q_1, q_3)
13. (q_1, q_3q_4)
14. (q_1, q_2)
15. (q_1, q_2q_3)
16. (q_1, q_2q_4)
17. $(q_1, q_2q_3q_4)$
18. (q_4, q_1q_2)
19. (q_3, q_1q_2)
20. (q_1q_2, q_3q_4)
21. (q_4, q_1q_3)
22. (q_2, q_1q_3)
23. (q_1q_3, q_2q_4)
24. $(q_4, q_1q_2q_3)$
25. (q_3, q_1q_4)
26. (q_2, q_1q_4)
27. (q_1q_4, q_2q_3)
28. $(q_3, q_1q_2q_4)$
29. $(q_2, q_1q_3q_4)$



e change the order?

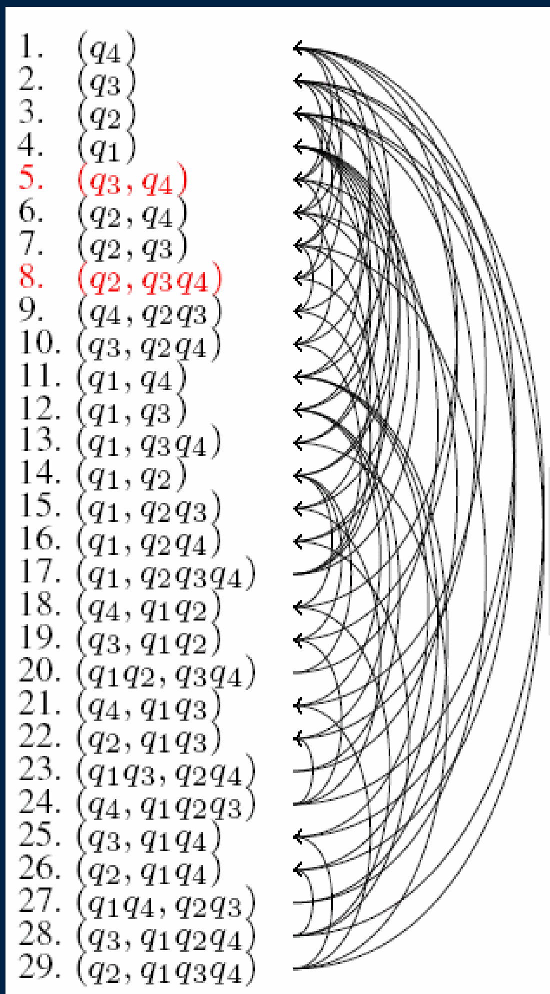
Problem Statement

- Develop a *generic* framework for parallelizing *any* bottom-up query optimizer

Overview of Solution

- A totally ordered sequence of pairs of quantifier sets is generated in a streaming fashion
- Buffer a fixed number of pairs and delay plan generation for those pairs
- Perform dependency-aware reordering
  - Convert the total order into a partial order over groups
  - Execute *parallel group topological sort*
- To maximize parallelism, we propose three novel optimizations

A Partial Order



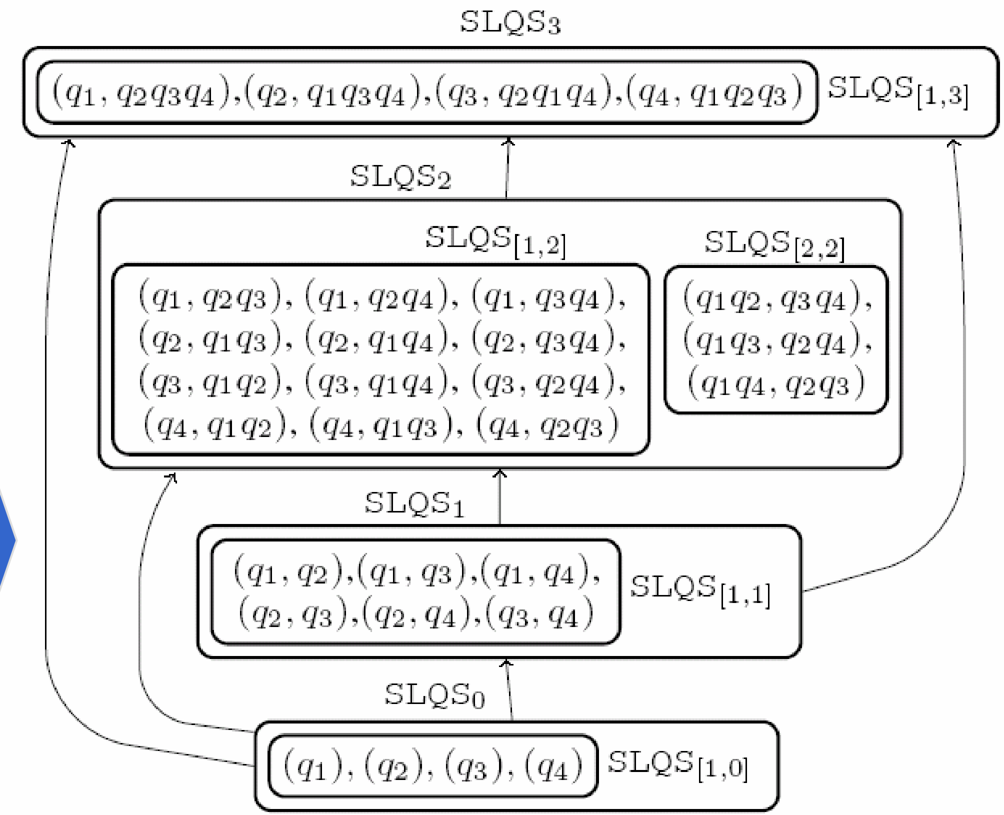
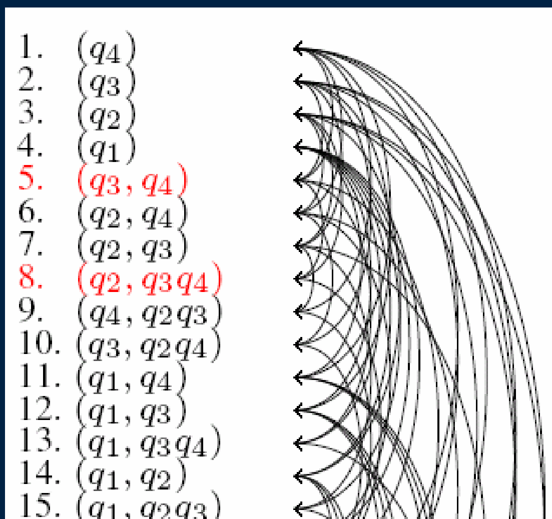


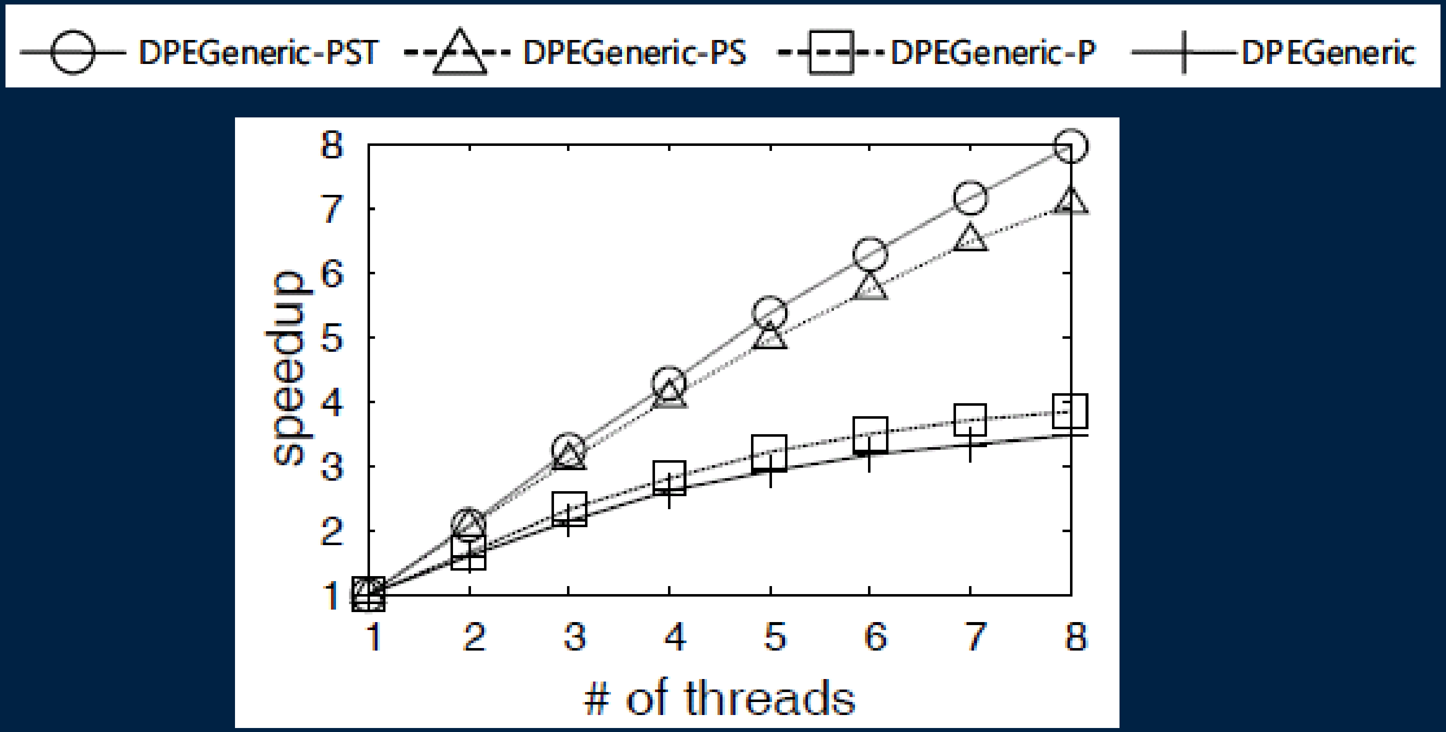
Figure 4: An example of partial order $\preceq_{SLQS}$.

Example


1. (q_4)
2. (q_3)
3. (q_2)
4. (q_1)
5. (q_3, q_4)
6. (q_2, q_4)
7. (q_2, q_3)
8. $(q_2, q_3 q_4)$
9. $(q_4, q_2 q_3)$
10. $(q_3, q_2 q_4)$
11. (q_1, q_4)
12. (q_1, q_3)
13. $(q_1, q_3 q_4)$
14. (q_1, q_2)
15. $(q_1, q_2 q_3)$

Experiment 1

- To determine how much the three optimization techniques contribute to maximize the parallelism of DPEGeneric



Star query with 20 quantifiers

THANK YOU!

Any Question?