

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]

- 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

The screenshot shows the IMDb website for the movie 'The Godfather (1972)'. The browser window title is 'IMDb - Google 검색' and the address bar shows 'http://www.imdb.com/title/tt0068646/'. The page features a navigation bar with links for 'NOW PLAYING', 'MOVIE/TV NEWS', 'MY MOVIES', 'DVD & BLU-RAY', 'IMDb TV', 'MESSAGE BOARDS', 'SHOWTIMES & TICKETS', 'IMDbPRO', and 'IMDb Resume'. The main content area includes a search bar, a 'The Godfather (1972)' header, a 'Photos' section with a grid of images, and an 'Overview' section. The 'Overview' section displays a user rating of 9.1/10 based on 357,064 votes, a MOVIEmeter showing a 5% increase in popularity, and the director 'Francis Ford Coppola' highlighted with a red box. Other details include the release date (25 May 1977), genre (Crime | Drama | Thriller), and a plot summary. A 'classmates.com' advertisement is visible on the right side of the page.

IMDb > The Godfather (1972)

The Godfather (1972) [More at IMDbPro »](#)

Photos ([see all 187](#) | [slideshow](#))

Overview

User Rating: 9.1/10 [357,064 votes](#)
[Top 250: #2](#) ([register to vote](#))

MOVIEmeter: Up 5% in popularity this week. See [rank & trends](#) on IMDbPro.

Director: **Francis Ford Coppola**

Writers: [Mario Puzo](#) (novel)
[Mario Puzo](#) (screenplay) ... [more](#)

Contact: View [company](#) contact information for The Godfather on [IMDbPro](#).

Release Date: 25 May 1977 (South Korea) [more](#)

Genre: [Crime](#) | [Drama](#) | [Thriller](#) [more](#)

Tagline: An offer you can't refuse.

Plot: The aging patriarch of an organized crime dynasty transfers control of his clandestine empire to his reluctant son. [full summary](#) | [full synopsis](#)

Plot Keywords: Spoiler alert! Rollover or vote to view plot keywords [more](#)

Awards: Won 3 Oscars. Another 19 wins & 17 nominations [more](#)

[Watch it at Amazon](#)
[Buy it at Amazon](#)
[Rent it at blockbuster.com](#)

BETA

[Discuss in Boards](#)
[More at IMDb Pro](#)
[Add to My Movies](#)
[Update Data](#)

Quicklinks
[main details](#)

Top Links
[trailers and videos](#)
[full cast and crew](#)
[trivia](#)
- official sites
[memorable quotes](#)

Overview
[main details](#)
[combined details](#)
[full cast and crew](#)

I graduated in
classmates.com

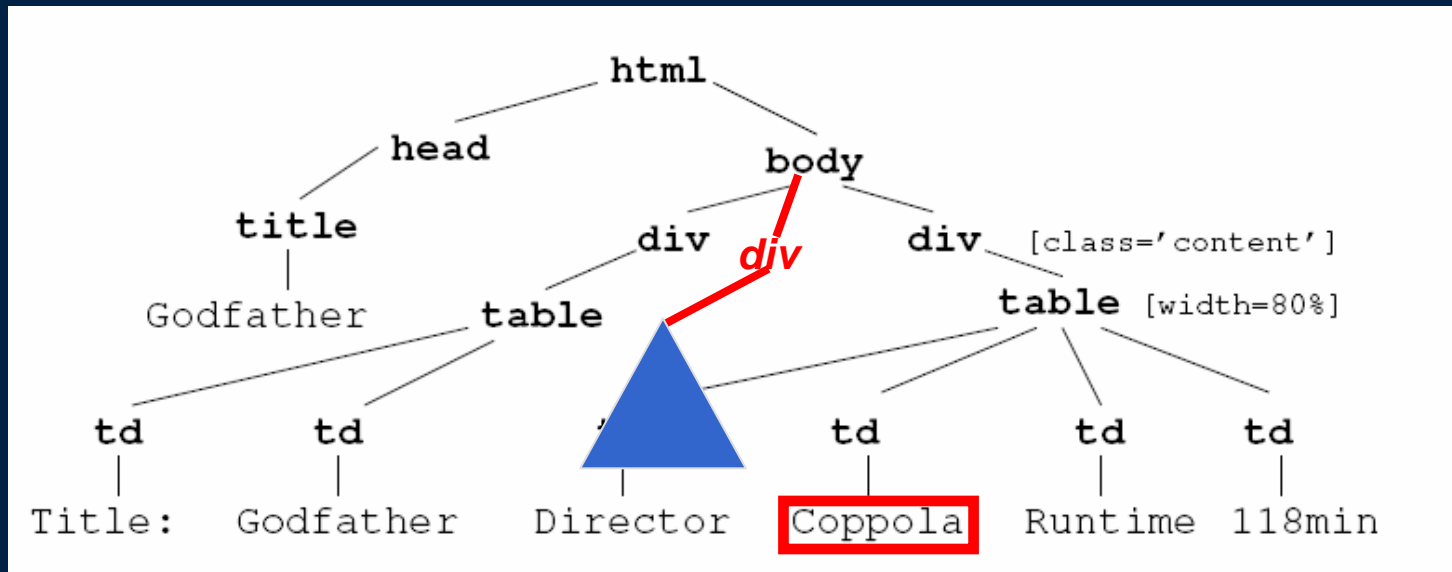
AL AK AZ AR CA CO CT DE DC FL GA HI ID IL IN IA KS KY LA ME MD MA MI MN MS MO MT NE NV NH NJ NM NY NC ND OH OK OR PA RI SC SD TN TX UT VT VA WA WV WI WY

I graduated in
classmates.com

AL AK AZ AR CA CO CT DE DC FL GA HI ID IL IN IA KS KY LA ME MD MA MI MN MS MO MT NE NV NH NJ NM NY NC ND OH OK OR PA RI SC SD TN TX UT VT VA WA WV WI WY

advertisement

A Web Page at Time T_0



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

Motivation

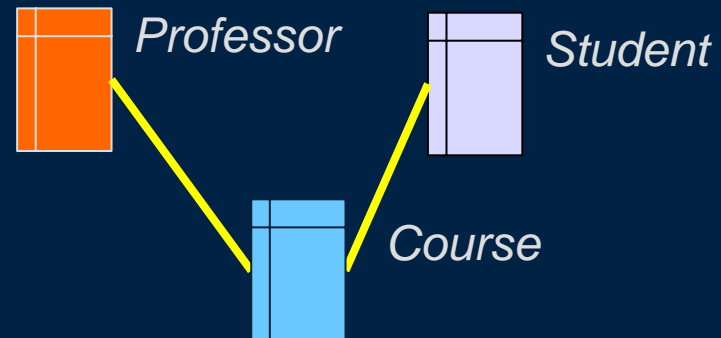
- The current state-of-the-art extraction systems are *very vulnerable* to small changes on the web
- A robust extraction solution is needed!

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
```

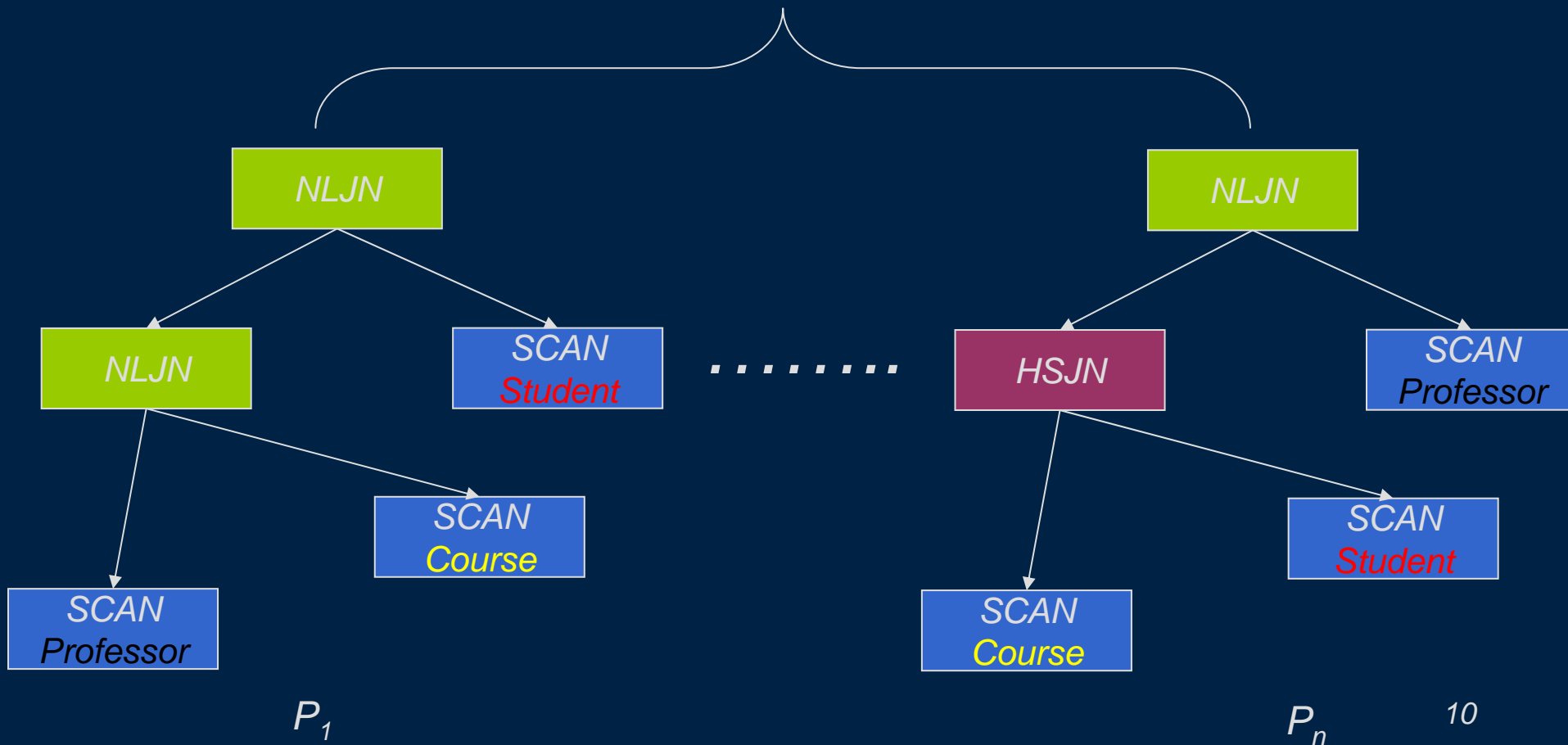


*Only need to specify **what** to find!*

Query Execution Plans (QEPs)

Plan = Query Execution Plan (QEP) = Access Plan

Generate the same result for the query!



Example: Query Optimization

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

quantifier



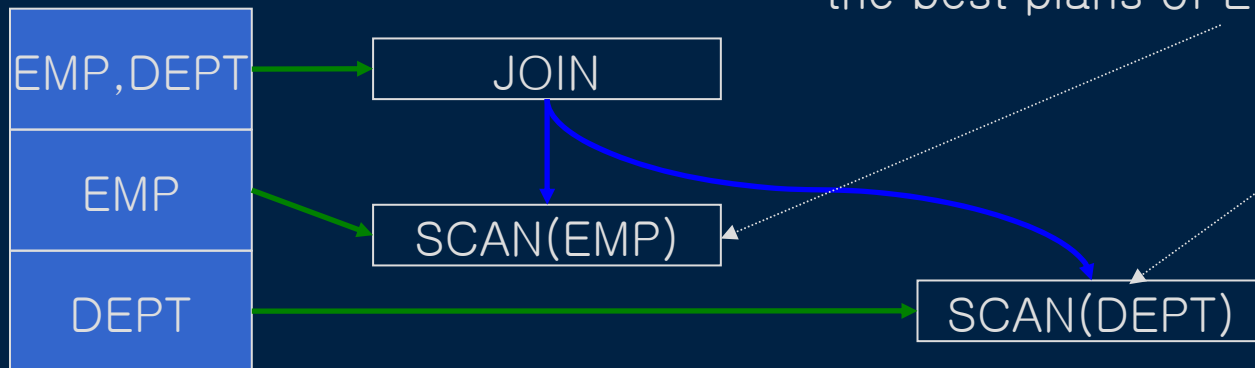
Level 1:

- 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

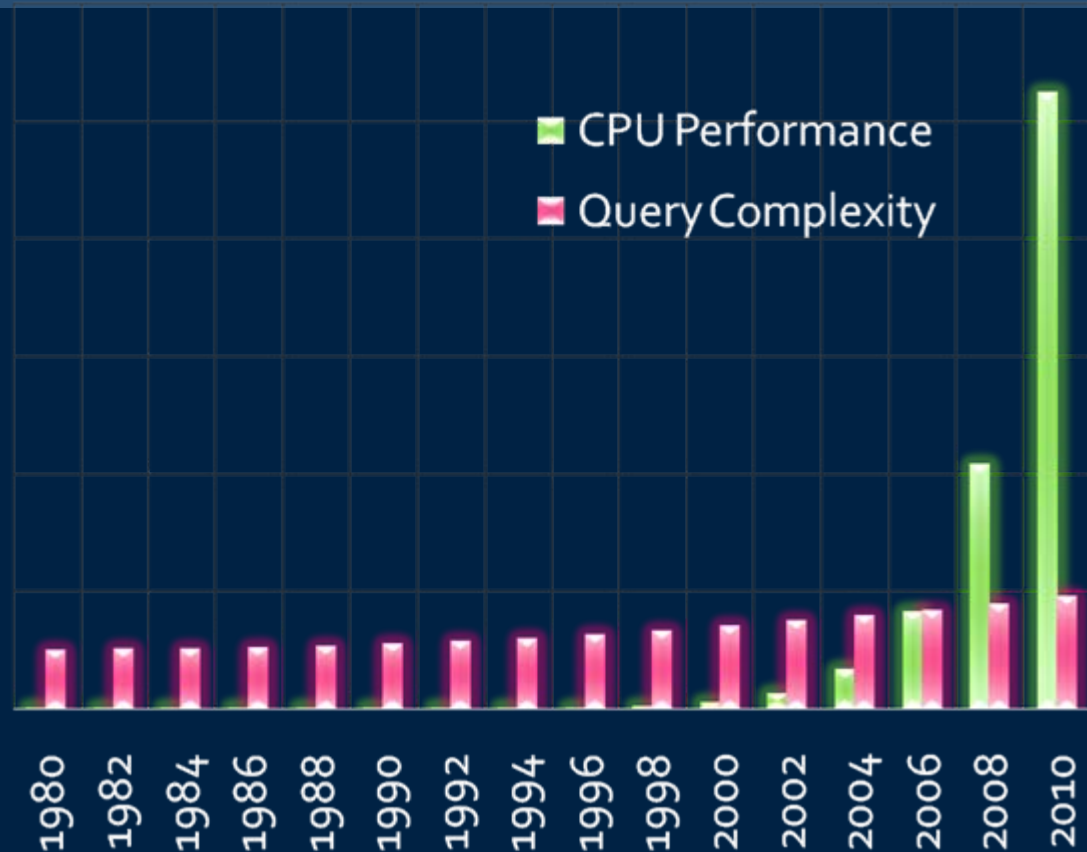
MEMO table (HASH table)



DP Query Optimization

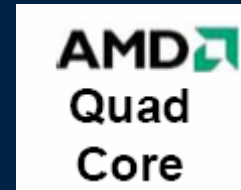
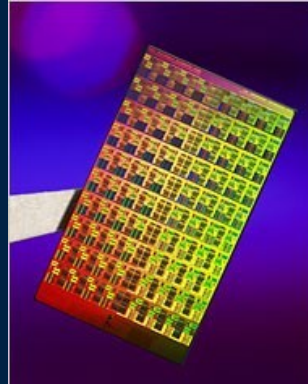
- Query optimization exploits dynamic programming (DP) to avoid generating redundant QEPs.
- Query optimization times using DP can increase significantly as the number of joins in a query increases.

Query Complexity vs. CPU Performance



Moore's Law outperforms query complexity

Multi-Core Wave



Supporting up to **256** cores

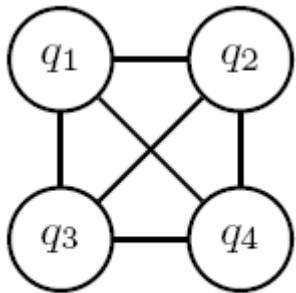
Motivation

- 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

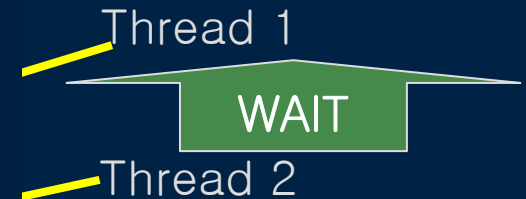
```

SELECT *
FROM R1 q1, R2 q2,
     R3 q3, R4 q4
WHERE q1.a2 = q2.a1 and
      q1.a3 = q3.a1 and
      q1.a4 = q4.a1 and
      q2.a3 = q3.a2 and
      q2.a4 = q4.a2 and
      q3.a4 = q4.a3
    
```



Query graph G

1. (q₄)
2. (q₃)
3. (q₂)
4. (q₁)
5. (q₃, q₄)
6. (q₂, q₄)
7. (q₂, q₃)
8. (q₂, q₃q₄)
9. (q₄, q₂q₃)
10. (q₃, q₂q₄)
11. (q₁, q₄)
12. (q₁, q₃)
13. (q₁, q₃q₄)
14. (q₁, q₂)
15. (q₁, q₂q₃)
16. (q₁, q₂q₄)
17. (q₁, q₂q₃q₄)
18. (q₄, q₁q₂)
19. (q₃, q₁q₂)
20. (q₁q₂, q₃q₄)
21. (q₄, q₁q₃)
22. (q₂, q₁q₃)
23. (q₁q₃, q₂q₄)
24. (q₄, q₁q₂q₃)
25. (q₃, q₁q₄)
26. (q₂, q₁q₄)
27. (q₁q₄, q₂q₃)
28. (q₃, q₁q₂q₄)
29. (q₂, q₁q₃q₄)



Can we 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

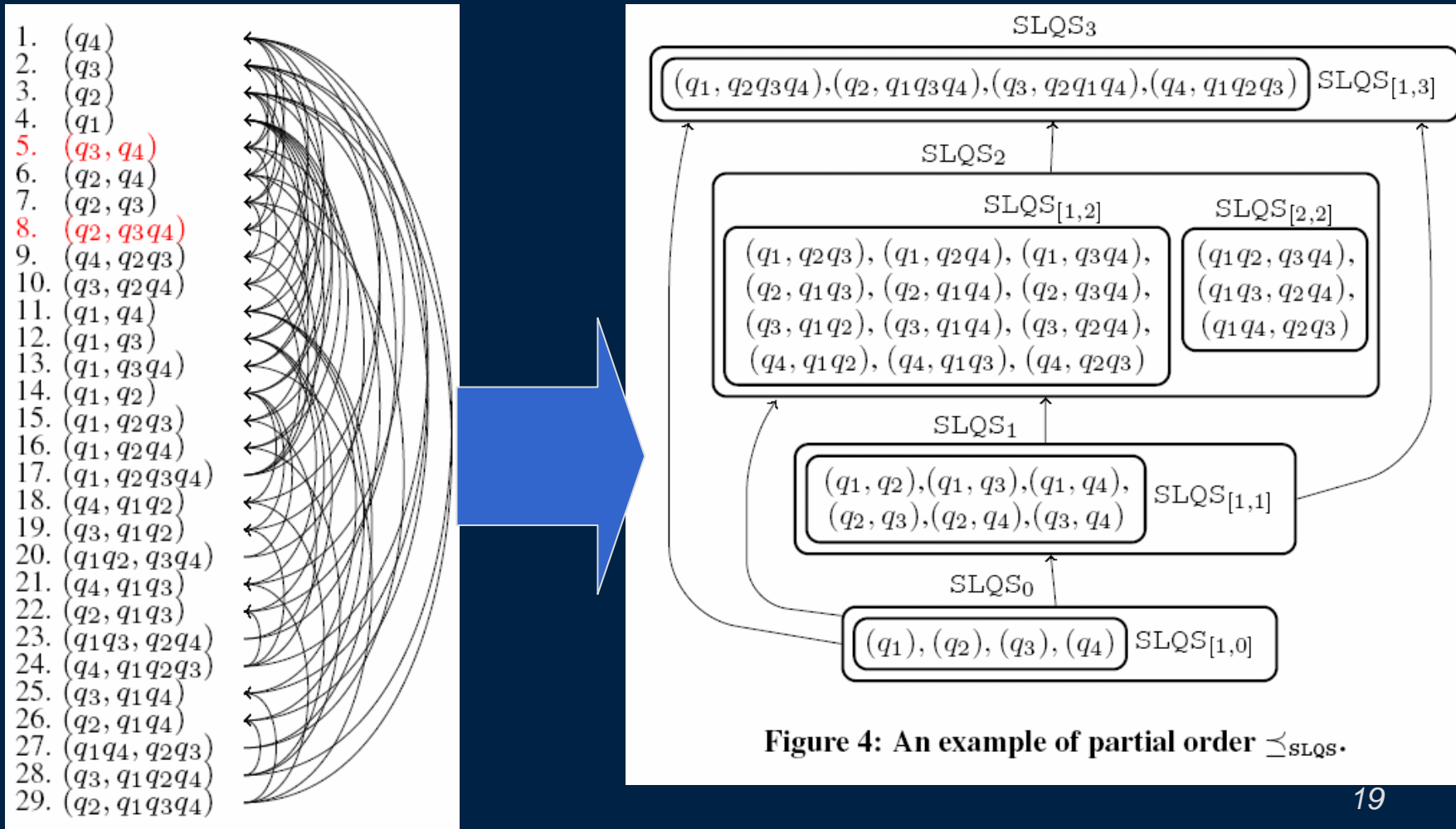
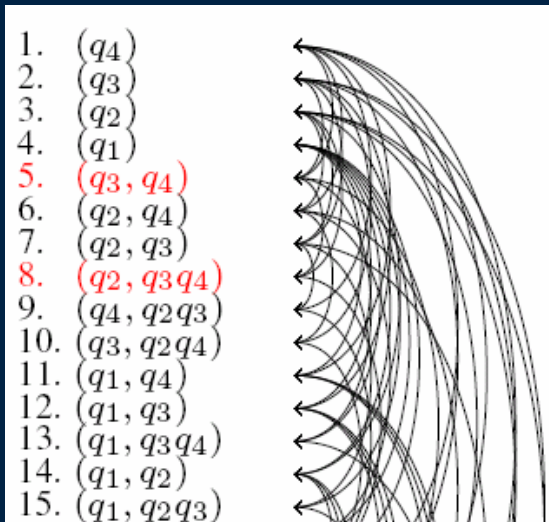


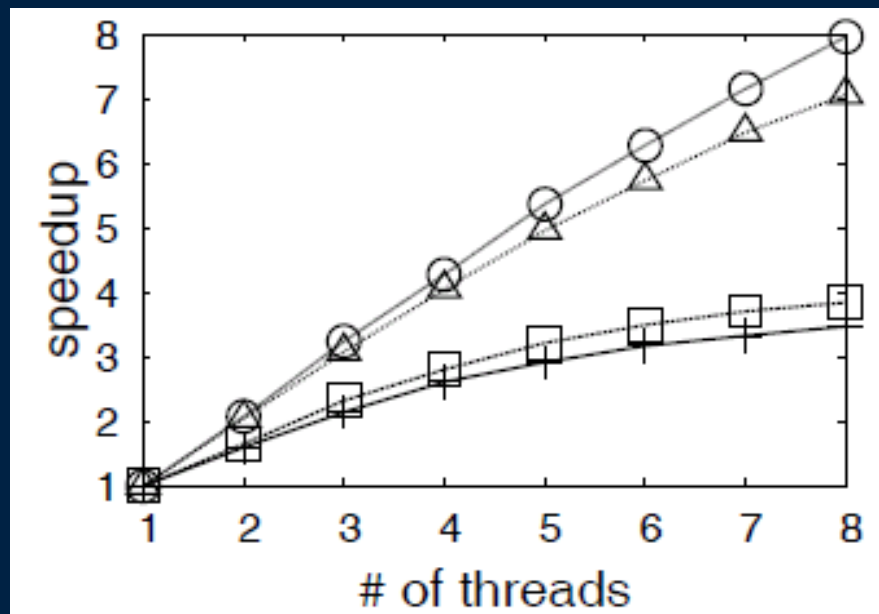
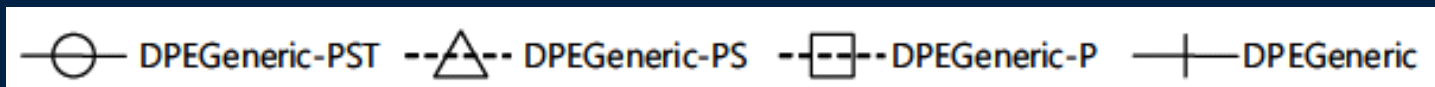
Figure 4: An example of partial order \preceq_{SLQS} .

Example



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?