ORACLE®

PGX.ISO: Parallel and Efficient In-Memory Engine for Subgraph Isomorphism

Raghavan Raman , PhD
Oskar van Rest
Sungpack Hong, PhD
Zhe Wu, PhD
Hassan Chafi*, PhD
Jay Banerjee, PhD

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

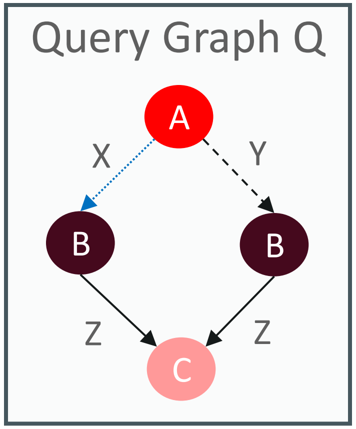
The Subgraph Isomorphism Problem

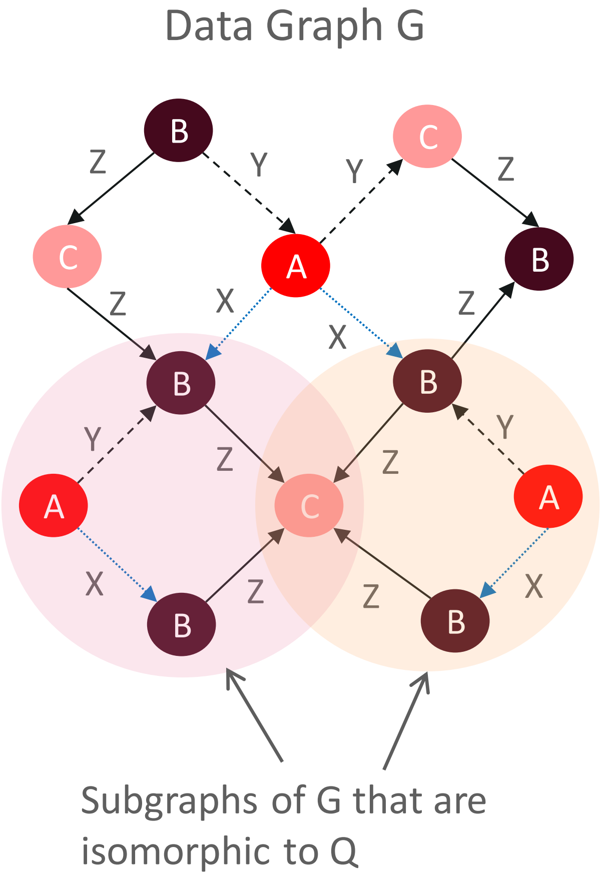
Q: Query Graph
G: Data Graph
A, B, C – Node Properties
X, Y, Z – Edge Properties

Problem: Find all subgraphs of G that are isomorphic to Q

Matching Criteria:

1. Topology of the graphs and
2. Properties on nodes and edges





Existing Solutions

- Graph Databases
  - RDF data Model
    - Oracle, Virtuoso, …
    - SPARQL: standard query language
  - Property Graph (PG) data model
    - Neo4J, …
    - No standard query language yet

➔ Disk-based solutions

- [pro] Process very large graphs
- [con] Disk latency becomes performance bottleneck

- *In-Memory* Solutions
  - Mostly from academia
    - VF2, QuickSI, TurboISO, etc…
  - Mostly sequential algorithms
  - Common approach:

backtracking + filtering ➔ prune partial solutions

[Issues and Lessons]

- Parallelizing backtracking algorithm s challenging – esp. load balancing
- Poor spatial locality from depth-first approaches
- Matching Order is important
- Need efficient partial solutions handling

ORACLE®

Our Approach (1) : PGX.ISO

- Parallel, In-memory engine for subgraph isomorphism
  - Use efficient data structure for graph and partial solutions
  - Considers load balancing and workload distribution
- Breadth-first search
  - Fixed order of query nodes for matching
  - Better for parallelization and more cache friendly
- Other optimizations
  - Different matching strategies for different graph patterns
  - Edge-first matching to improve performance

→ Visit our poster for details

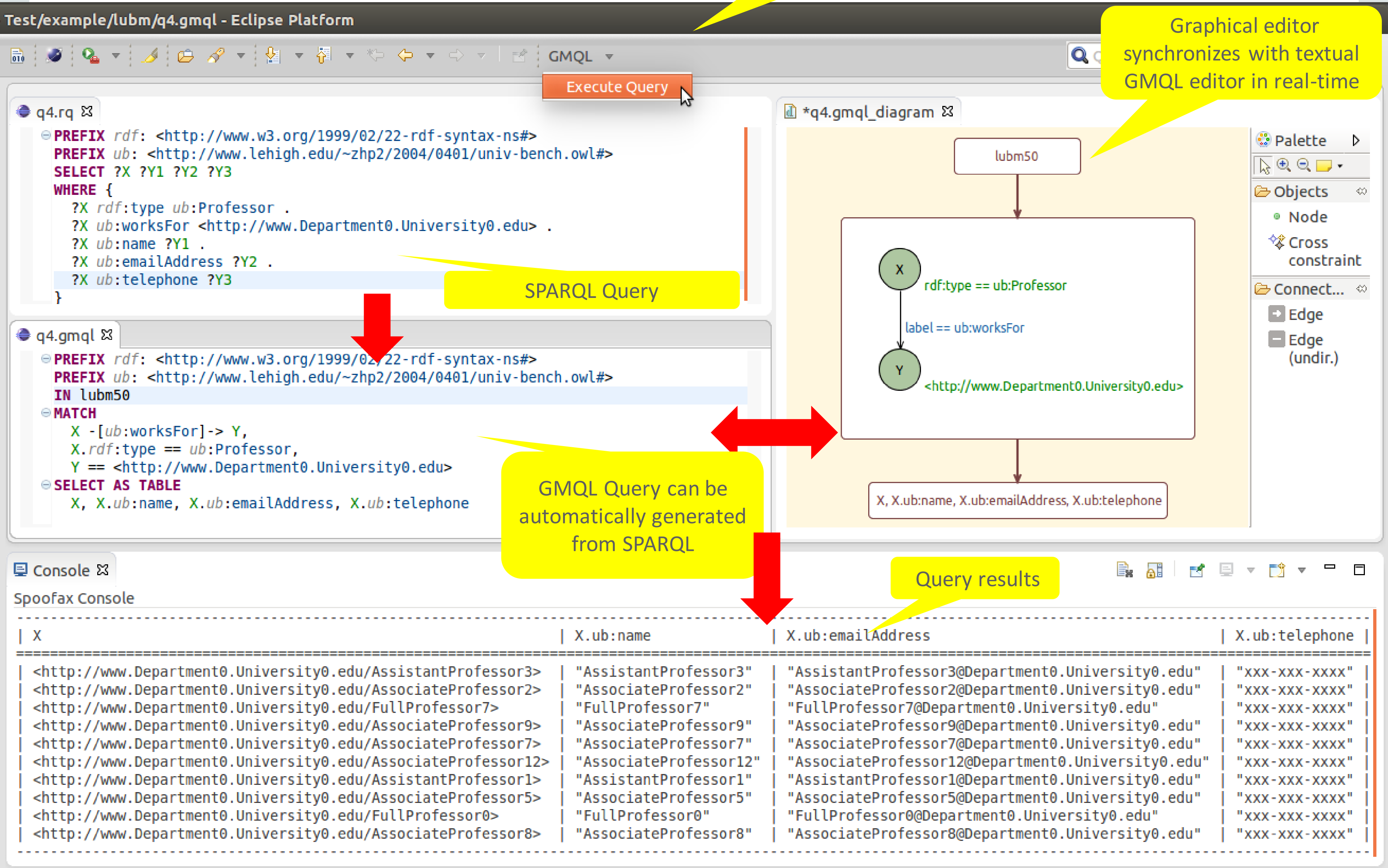
Our Approach (2) : GMQL

- GMQL: Graph Matching Query Language
  - A Query Language for *Property Graph* Data Model
  - First-class constructs for nodes, edges and properties
  - Compiles query into PGX.ISO
- Native SPARQL support
  - Automatic conversion: SPARQL → GMQL
- IDE and Visualization
  - Pluggable to Eclipse
  - Visualize query (and result)
  - Built from Spoofax language bench (TU DELFT)

→ Visit our poster for details

ORACLE®

GMQL: Look and Feel



Integrated with Eclipse

Graphical editor synchronizes with textual GMQL editor in real-time

SPARQL Query

GMQL Query can be automatically generated from SPARQL

Query results

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ub: <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#>
SELECT ?X ?Y1 ?Y2 ?Y3
WHERE {
  ?X rdf:type ub:Professor .
  ?X ub:worksFor <http://www.Department0.University0.edu> .
  ?X ub:name ?Y1 .
  ?X ub:emailAddress ?Y2 .
  ?X ub:telephone ?Y3
}
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ub: <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#>
IN lubm50
MATCH
  X -[ub:worksFor]-> Y,
  X.rdf:type == ub:Professor,
  Y == <http://www.Department0.University0.edu>
SELECT AS TABLE
  X, X.ub:name, X.ub:emailAddress, X.ub:telephone
```

```
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------
| X                                                              | X.ub:name              | X.ub:emailAddress                                      | X.ub:telephone |
=============================================================================================================================================================================
| <http://www.Department0.University0.edu/AssistantProfessor3>   | "AssistantProfessor3"  | "AssistantProfessor3@Department0.University0.edu"      | "xxx-xxx-xxxx" |
| <http://www.Department0.University0.edu/AssociateProfessor2>   | "AssociateProfessor2"  | "AssociateProfessor2@Department0.University0.edu"      | "xxx-xxx-xxxx" |
| <http://www.Department0.University0.edu/FullProfessor7>        | "FullProfessor7"       | "FullProfessor7@Department0.University0.edu"           | "xxx-xxx-xxxx" |
| <http://www.Department0.University0.edu/AssociateProfessor9>   | "AssociateProfessor9"  | "AssociateProfessor9@Department0.University0.edu"      | "xxx-xxx-xxxx" |
| <http://www.Department0.University0.edu/AssociateProfessor7>   | "AssociateProfessor7"  | "AssociateProfessor7@Department0.University0.edu"      | "xxx-xxx-xxxx" |
| <http://www.Department0.University0.edu/AssociateProfessor12>  | "AssociateProfessor12" | "AssociateProfessor12@Department0.University0.edu"     | "xxx-xxx-xxxx" |
| <http://www.Department0.University0.edu/AssistantProfessor1>   | "AssistantProfessor1"  | "AssistantProfessor1@Department0.University0.edu"      | "xxx-xxx-xxxx" |
| <http://www.Department0.University0.edu/AssociateProfessor5>   | "AssociateProfessor5"  | "AssociateProfessor5@Department0.University0.edu"      | "xxx-xxx-xxxx" |
| <http://www.Department0.University0.edu/FullProfessor0>        | "FullProfessor0"       | "FullProfessor0@Department0.University0.edu"           | "xxx-xxx-xxxx" |
| <http://www.Department0.University0.edu/AssociateProfessor8>   | "AssociateProfessor8"  | "AssociateProfessor8@Department0.University0.edu"      | "xxx-xxx-xxxx" |
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------
```

Performance Evaluation: PGX.ISO

- Dataset : LUBM Lubm datasets evaluated:
  - A standard benchmark for RDF/SPARQL
  - Lubm 8K – 173.8 million nodes, 701.8 million edges
  - Lubm 25K – 543 million nodes, 2.1 billion edges
- Environments (x86 and SPARC)
  - X86: 2 x 8-Core Intel(R) Xeon(R) CPU E5-2660 @ 2.2 GHz (X3-2)
  - SPARC: 8 x 16-Core SPARC T5 processor @ 3.6 GHz
- Comparisons
  - Oracle SPARQL SQL with Oracle RDBMS 12.1.0.1
  - SPARQL SQL queries run directly on the Oracle RDBMS
  - Graph is loaded into memory before running SPARQL queries in PGX

Performance on LUBM Queries

LUBM Query	LUBM 8K	Execution Time on x86 (s)	
	#Solutions	SQL	PGX.ISO
Query 1	4	0	0
Query 2	2528	21.26	0.1
Query 3	6	0	0
Query 4	34	0	0
Query 5	719	0.02	0
Query 6	83557706	23.56	0.14
Query 7	67	0.01	0
Query 8	7790	0.23	0
Query 9	2178420	58	0.58
Query 10	4	0	0
Query 11	224	0.01	0
Query 12	15	0.14	0
Query 13	37118	1.15	0.03
Query 14	63400587	21.09	0.1

Focus on 4 queries

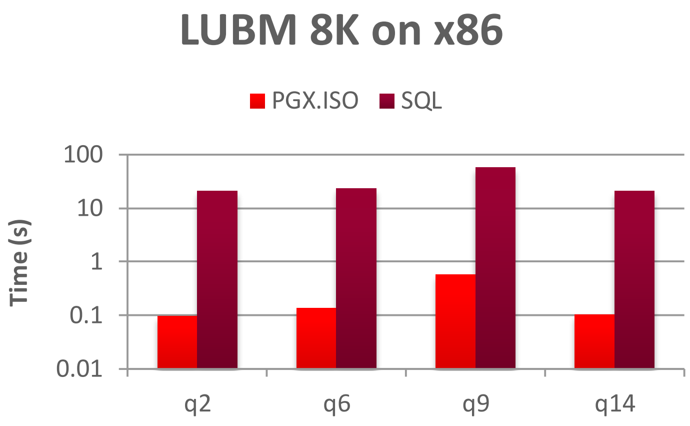
Time < 0.01 (s) is considered 0

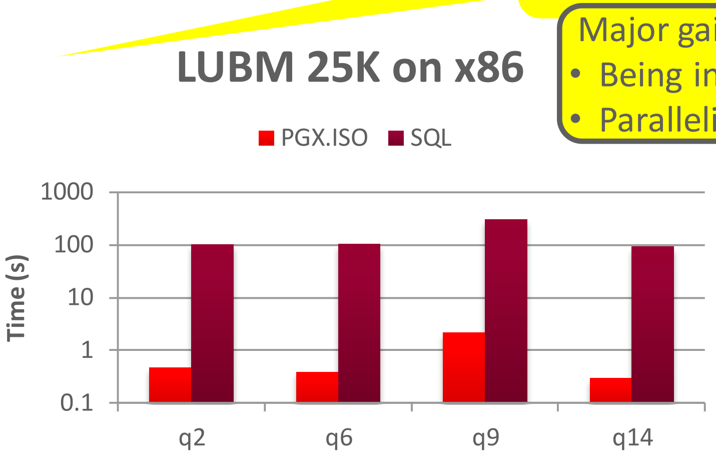
Comparison of PGX.ISO and Oracle-SQL

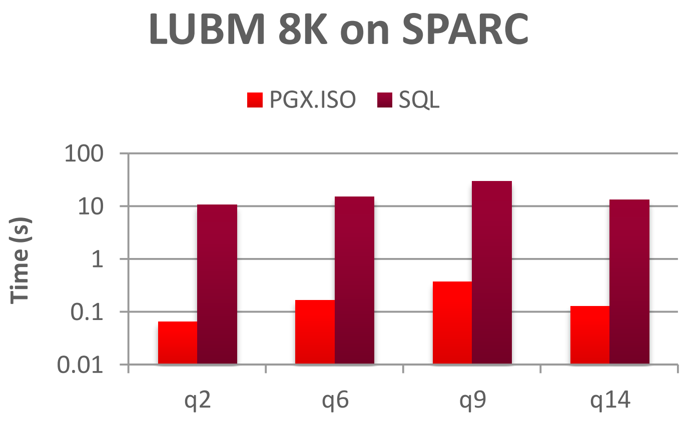
LUBM 8K and 25K on x86 and Sparc

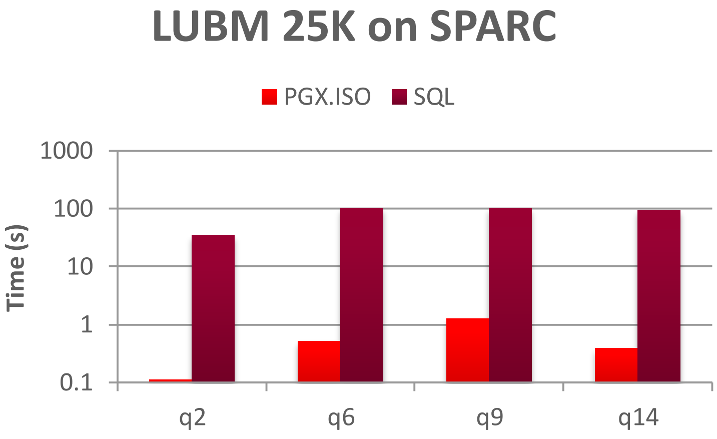
100x improvement over SQL for all queries

Major gains from :
- Being in-memory
- Parallelization


LUBM 8K on x86


LUBM 25K on x86

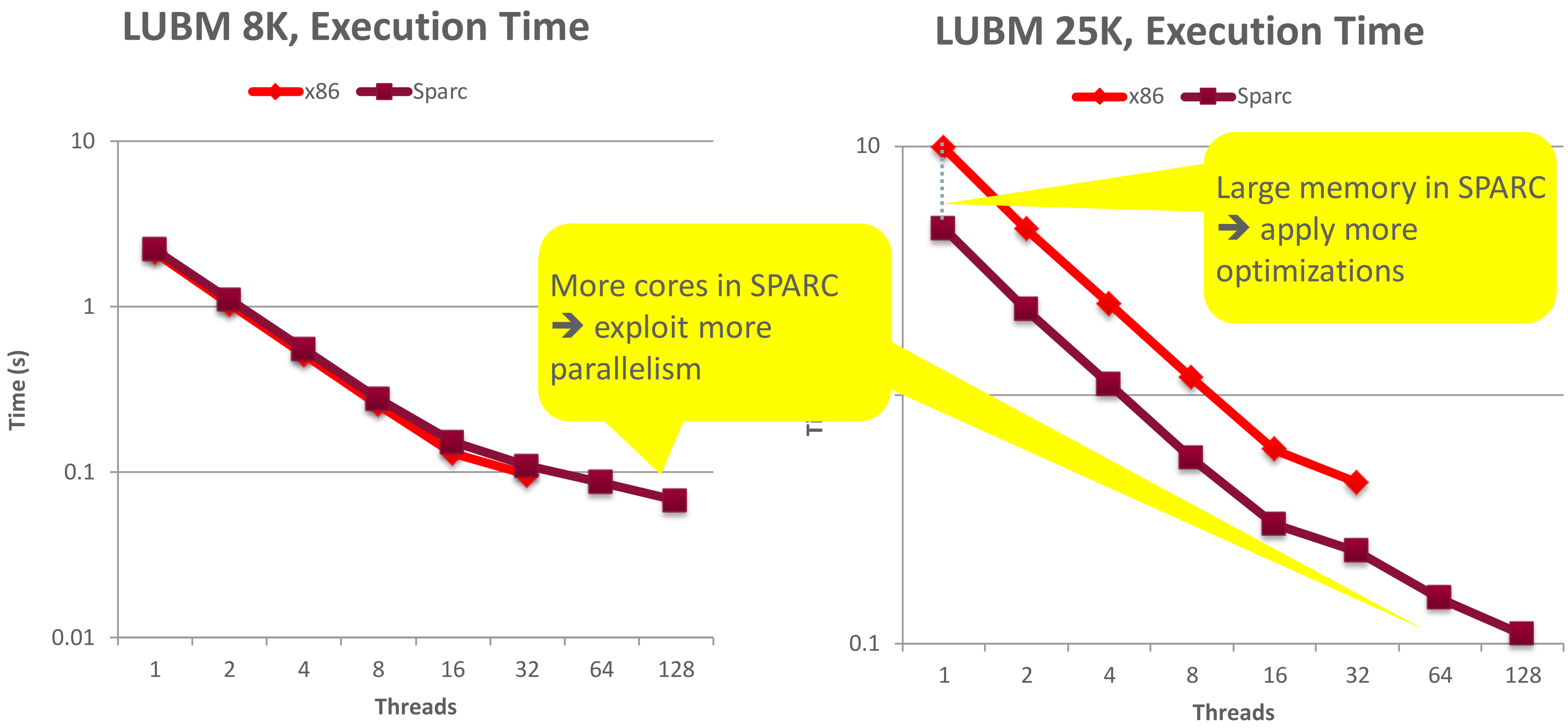

LUBM 8K on SPARC


LUBM 25K on SPARC

Scalability of PGX.ISO

LUBM Query 2 Scalability on x86 and SPARC

PGX.ISO is well-parallelized (for x86 and SPARC)

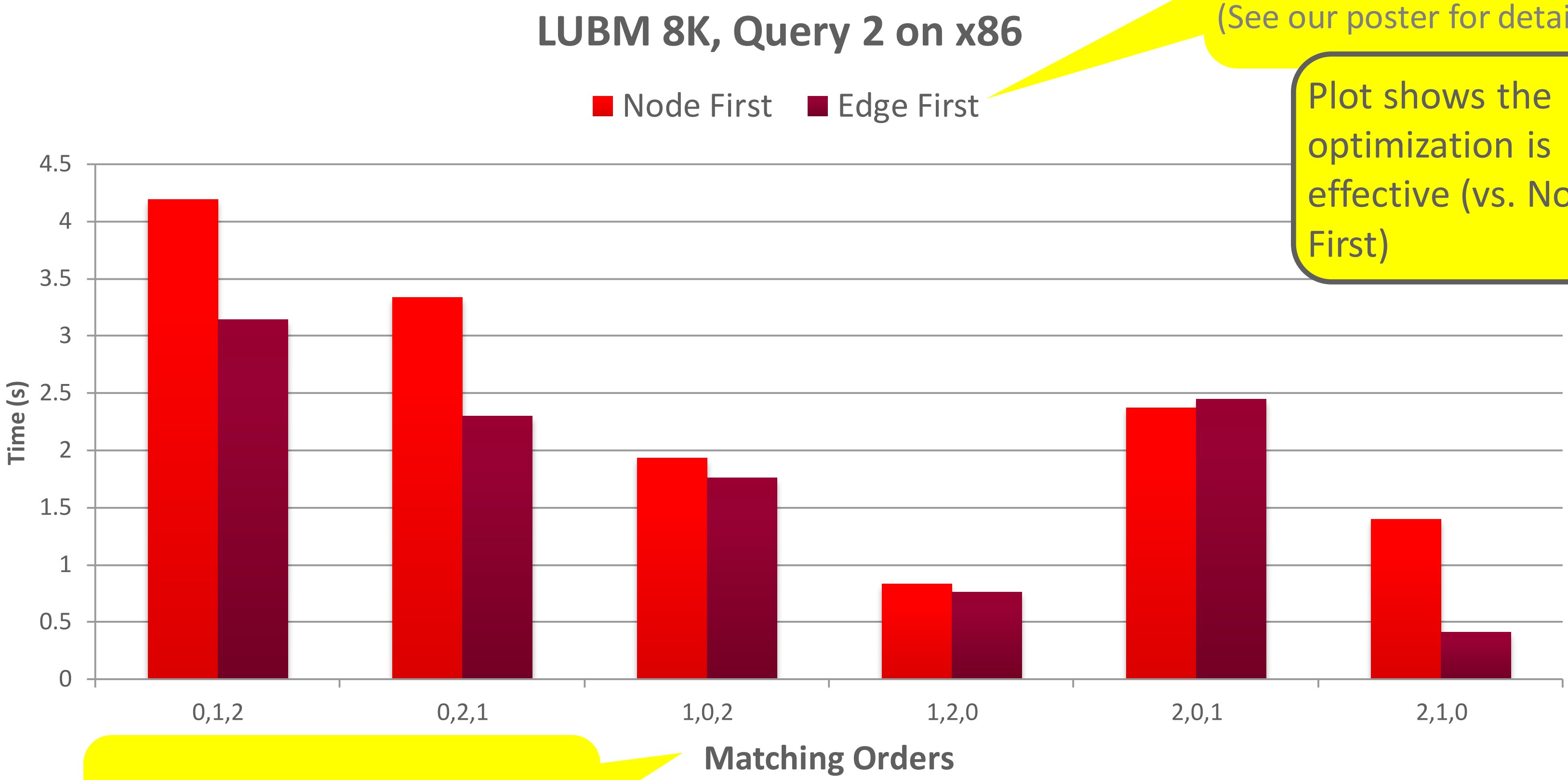


More cores in SPARC ➔ exploit more parallelism

Large memory in SPARC ➔ apply more optimizations

*Best numbers for x86 and SPARC (with different optimizations and matching orders)

Closer Look and Remaining Issues



Edge First: one of our optimization
(See our poster for details)

Plot shows the optimization is effective (vs. Node First)

[Observation] Performance depends on matching order

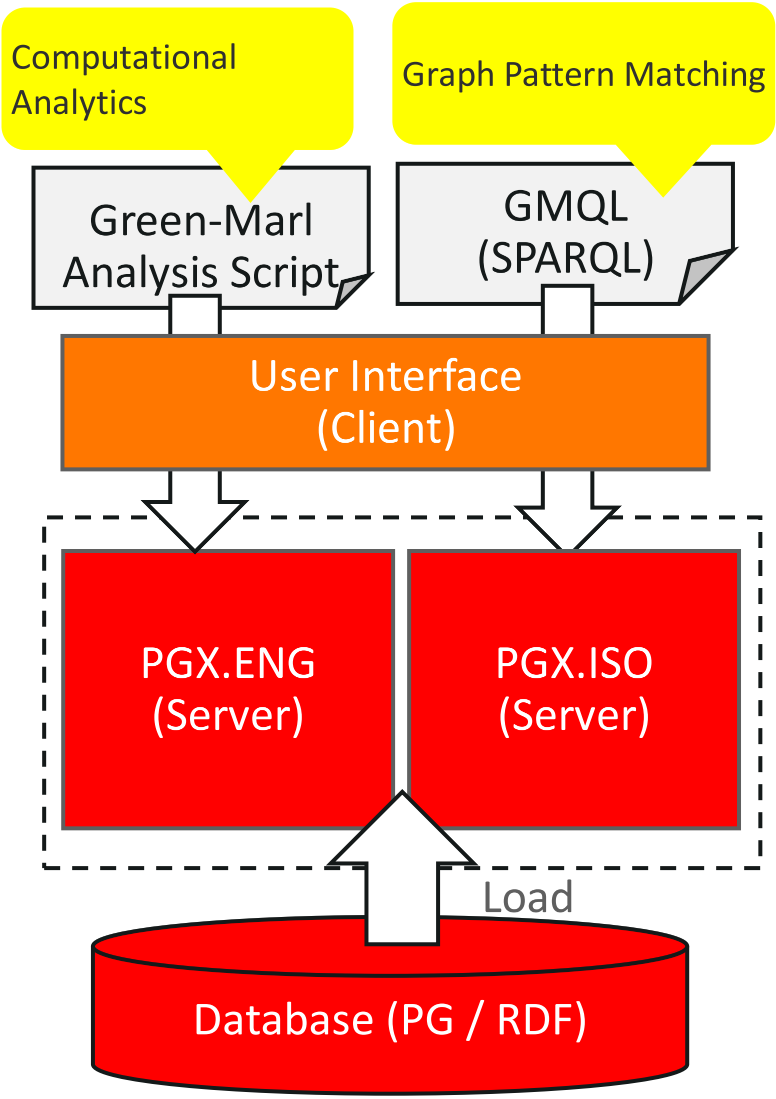
➔ On-going research: finding best matching order

The whole PGX System

- PGX
  - In-memory, parallel graph analytic engine
  - Use database as persistence layer
  - Load graph into memory
- Two kinds of workloads
  - Graph query (this paper)
  - ➔ Find patterns in graph
  - Computational analytics (OTN)
  - ➔ Page rank, community detection, …
  - We are merging these two engines

Check PGX engine at :

http://tinyurl.com/olabspgx



Summary

- PGX.ISO
  - Parallel, in-memory solution for subgraph isomorphism
- GMQL
  - a query language for property graph data
  - Provides RDF/SPARQL compatibility
- Evaluation with LUBM
  - With x86 and SPARC
  - Up to 300x faster than SQL-based Implementation

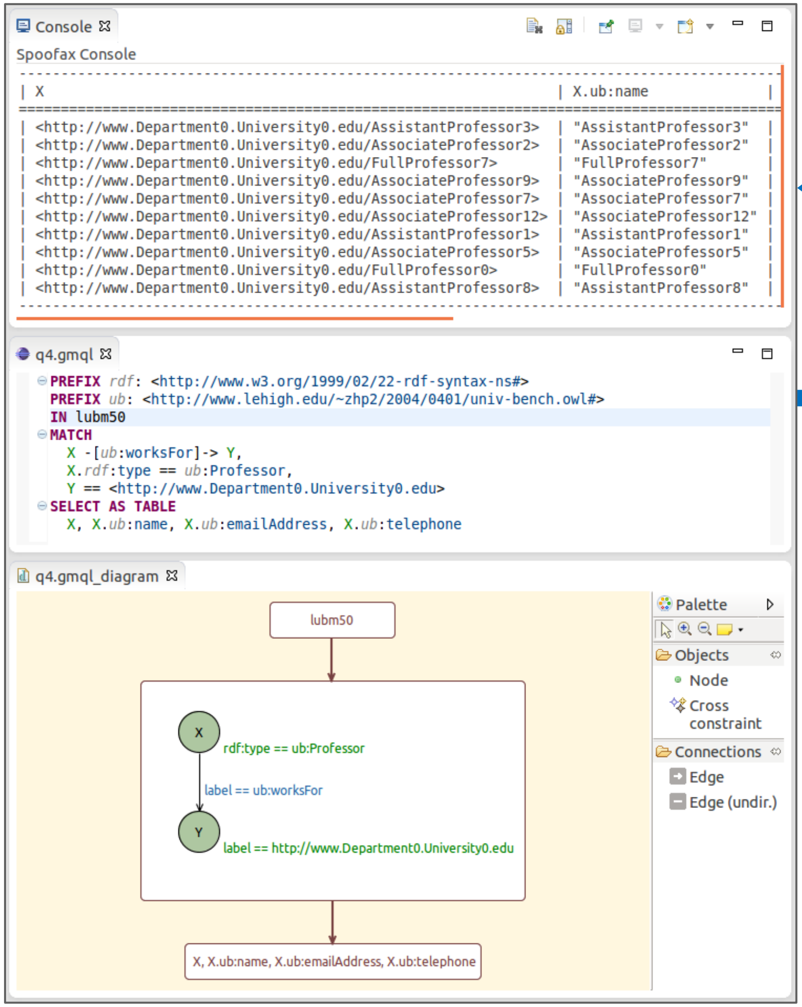
Hardware and Software

Engineered to Work Together

ORACLE®

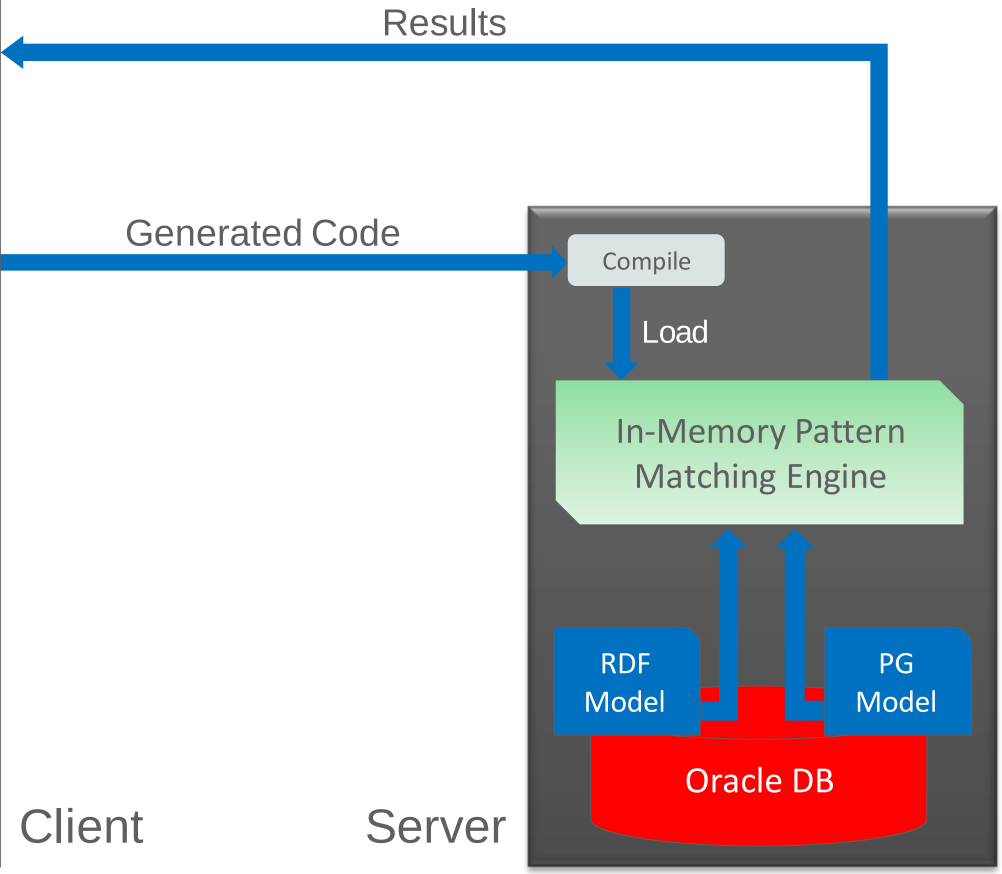
Backup Slides

GMQL: Graph-Matching Query Language



```
-------------------------------------------------------------------------------------------
| X                                                       | X.ub:name                 |
===========================================================================================
| <http://www.Department0.University0.edu/AssistantProfessor3>  | "AssistantProfessor3"  |
| <http://www.Department0.University0.edu/AssociateProfessor2>  | "AssociateProfessor2"  |
| <http://www.Department0.University0.edu/FullProfessor7>       | "FullProfessor7"       |
| <http://www.Department0.University0.edu/AssociateProfessor9>  | "AssociateProfessor9"  |
| <http://www.Department0.University0.edu/AssociateProfessor7>  | "AssociateProfessor7"  |
| <http://www.Department0.University0.edu/AssociateProfessor12> | "AssociateProfessor12" |
| <http://www.Department0.University0.edu/AssistantProfessor1>  | "AssistantProfessor1"  |
| <http://www.Department0.University0.edu/AssociateProfessor5>  | "AssociateProfessor5"  |
| <http://www.Department0.University0.edu/FullProfessor0>       | "FullProfessor0"       |
| <http://www.Department0.University0.edu/AssistantProfessor8>  | "AssistantProfessor8"  |
-------------------------------------------------------------------------------------------
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ub: <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#>
IN lubm50
MATCH
  X -[ub:worksFor]-> Y,
  X.rdf:type == ub:Professor,
  Y == <http://www.Department0.University0.edu>
SELECT AS TABLE
  X, X.ub:name, X.ub:emailAddress, X.ub:telephone
```


Results
Generated Code
Compile
Load
In-Memory Pattern Matching Engine
RDF Model
PG Model
Oracle DB
Client
Server