Graph Databases for Large-Scale Healthcare Systems

A Framework for Efficient Data Management and Data Services

Yubin Park[1], *Mallikarjun (Arjun) Shankar[2],* Byung-Hoon Park[2], and Joydeep Ghosh[1]

[1] University of Texas, Austin

[2] Oak Ridge National Laboratory

Yubin.park@utexas.edu, *shankarm@ornl.gov*

Databases in Healthcare

- Heterogeneous population
  - Physicians, Patients, Hospitals, Insurance Companies, and other healthcare agencies.
- Frequent changes on database structure
- Non-trivial entries
  - Set values: diagnosis and procedure
  - Natural language: doctors’ and nurses’ notes

Goals

- We investigate the usefulness of graph database in the healthcare domain
- We propose a simple set of transformation rules from 3NF RDBMS to graph database
- We compare the performance of the proposed graph database using synthetic data

3NF to Graph

Tables:
Primary keys
+ Foreign keys
+ Other attributes

Claim ID	Diagnosis	Procedure	Patient ID	Doctor ID
c1928	462.00	46.22	p1234	d1836
c6548	135.01	78.21	p5678	d4681
c9784	283.1	46.32	p1234	d1836
c3184	V30.01	21.00	p9183	d1836
c1083	193.91	81.00		

Claim table

Patient ID	Age	Gender	ZIP
p1234	75	Male	78751
p5678	62	Female	78793
p8910	43	Male	78704
p1112	86	Male	65264

Patient table

Doctor ID	Specialty	Telephone
d1836	75	512-983-xxxx
d4681	62	712-193-xxxx
d9187		
d9294		

Doctor table

3NF to Graph

Transform

Primary keys to “nodes”

Foreign key relationships to “edges”

The rest to “properties”

Claim ID	Diagnosis	Procedure	Patient ID	Doctor ID
c1928	462.00	46.22	p1234	d1836
c6548	135.01	78.21	p5678	d4681
c9784	283.1	46.32	p1234	d1836
c3184	V30.01	21.00	p9183	d1836
c1083	193.91	81.00		

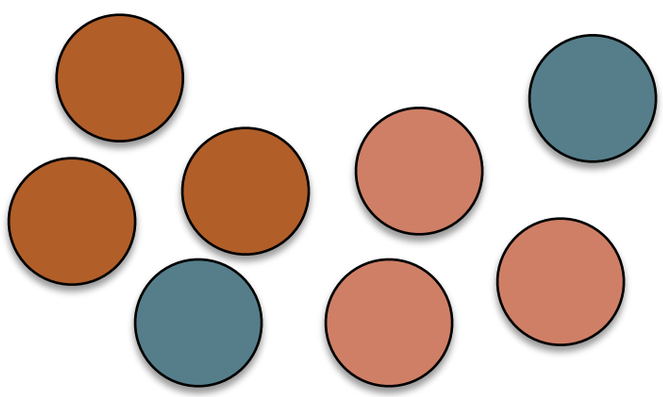
Claim table

Patient ID	Age	Gender	ZIP
p1234	75	Male	78751
p5678	62	Female	78793
p8910	43	Male	78704
p1112	86	Male	65264

Patient table

Doctor ID	Specialty	Telephone
d1836	75	512-983-xxxx
d4681	62	712-193-xxxx
d9187		
d9294		

Doctor table



3NF to Graph

Transform
Primary keys to “nodes”
Foreign key relationships to “edges”
The rest to “properties”

Claim ID	Diagnosis	Procedure	Patient ID	Doctor ID
c1928	462.00	46.22	p1234	d1836
c6548	135.01	78.21	p5678	d4681
c9784	283.1	46.32	p1234	d1836
c3184	V30.01	21.00	p9183	d1836
c1083	193.91	81.00		

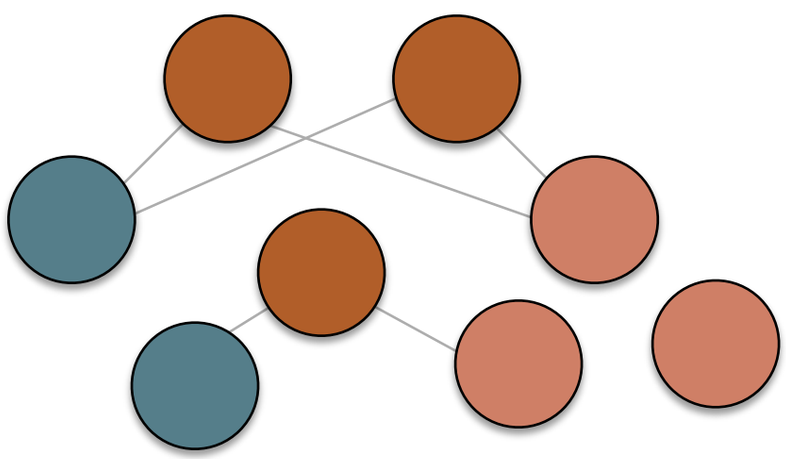
Claim table

Patient ID	Age	Gender	ZIP
p1234	75	Male	78751
p5678	62	Female	78793
p8910	43	Male	78704
p1112	86	Male	65264

Patient table

Doctor ID	Specialty	Telephone
d1836	75	512-983-xxxx
d4681	62	712-193-xxxx
d9187		
d9294		

Doctor table



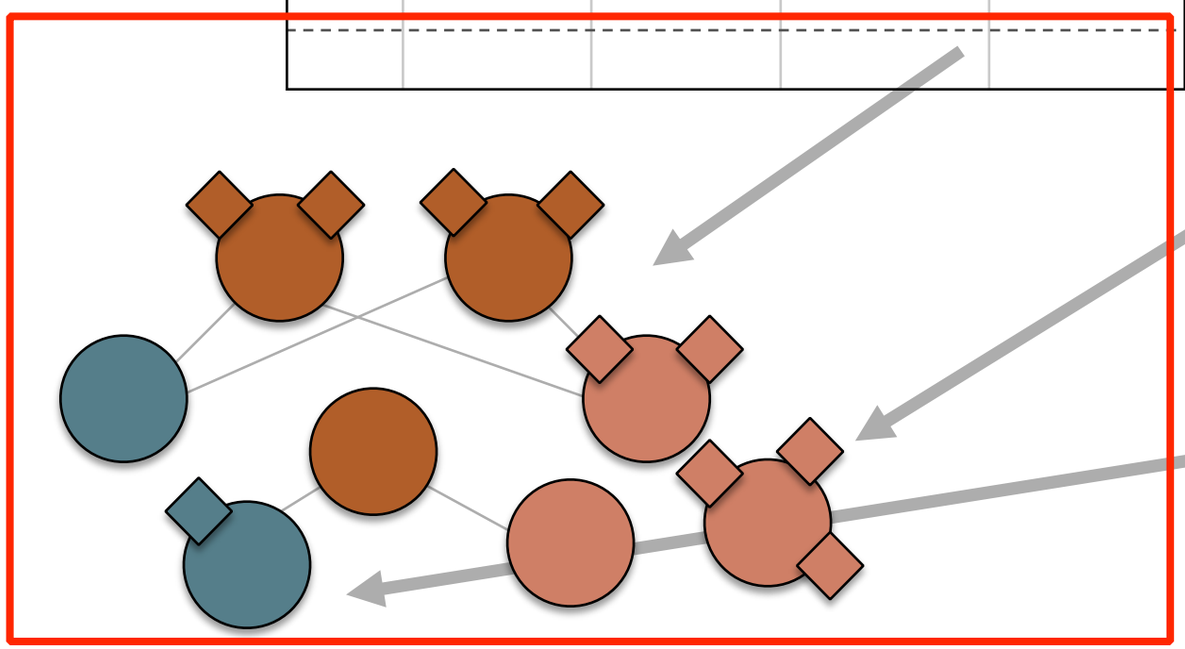
3NF to Graph

Transform
Primary keys to “nodes”
Foreign key relationships to “edges”
The rest to “properties”

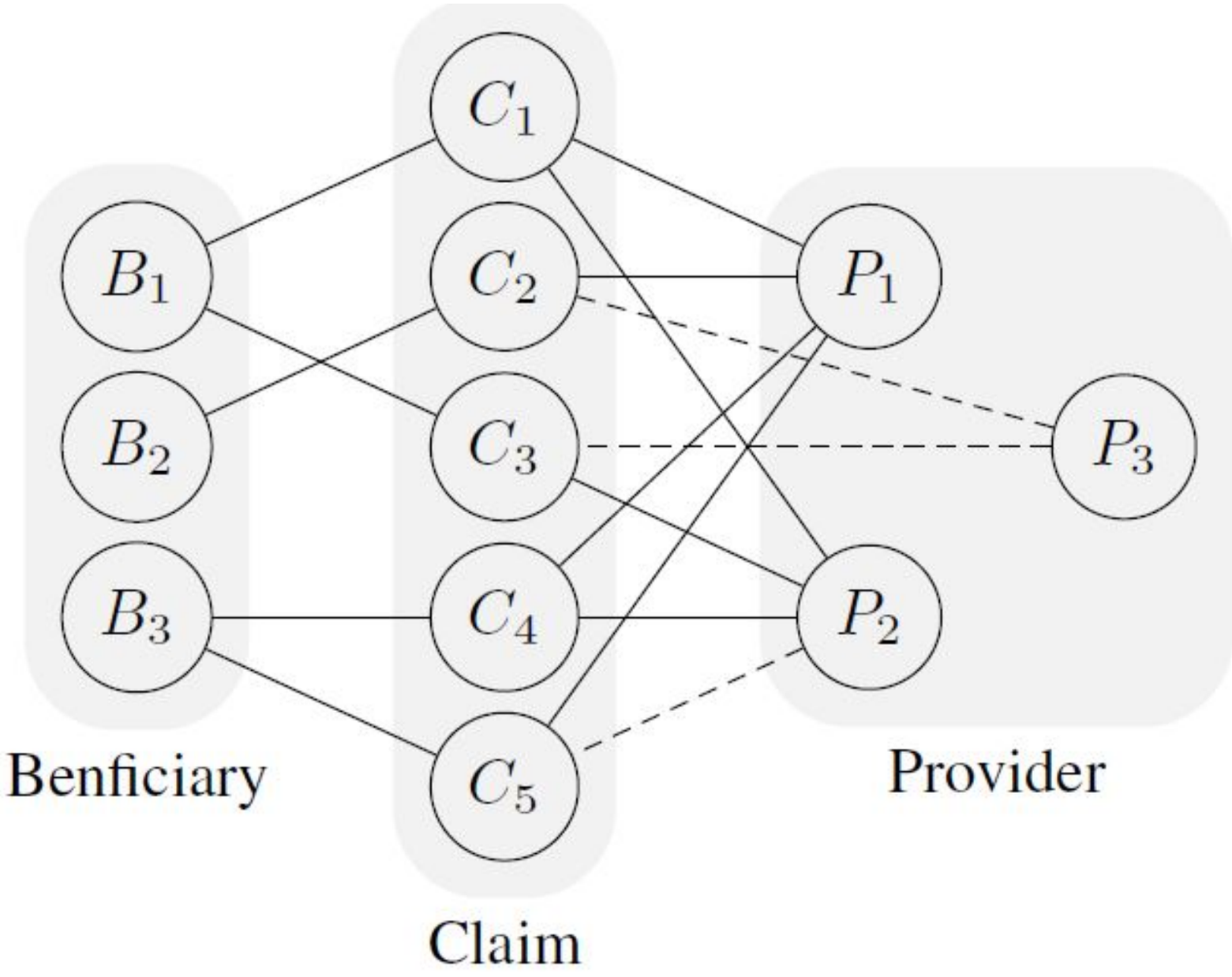
Claim ID	Diagnosis	Procedure	Patient ID	Doctor ID
c1928	462.00	46.22	p1234	d1836
c6548	135.01	78.21	p5678	d4681
c9784	283.1	46.32	p1234	d1836
c3184	V30.01	21.00	p9183	d1836
c1083	193.91	81.00		

Patient ID	Age	Gender	ZIP
p1234	75	Male	78751
p5678	62	Female	78793
p8910	43	Male	78704
p1112	86	Male	65264

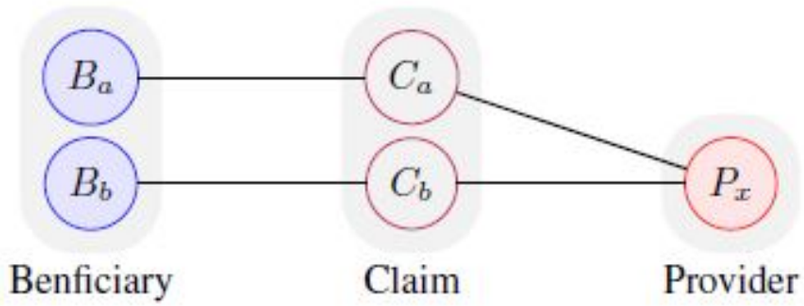
Doctor ID	Specialty	Telephone
d1836	75	512-983-xxxx
d4681	62	712-193-xxxx
d9187		
d9294		



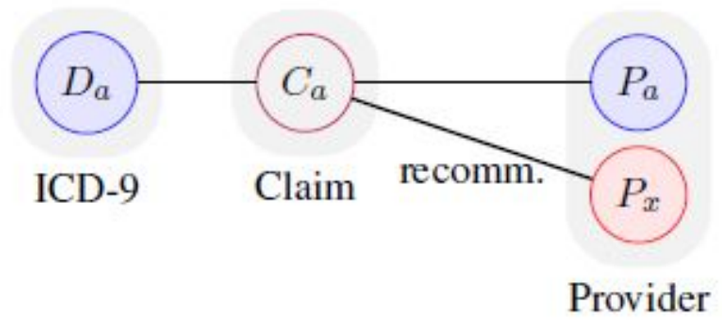
Graph Structure and Healthcare Entities



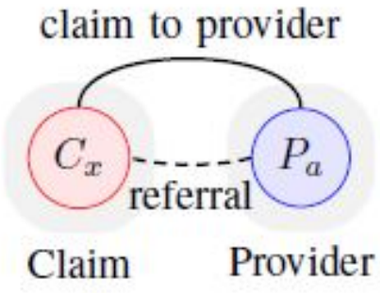
Graph Structure Interpretation



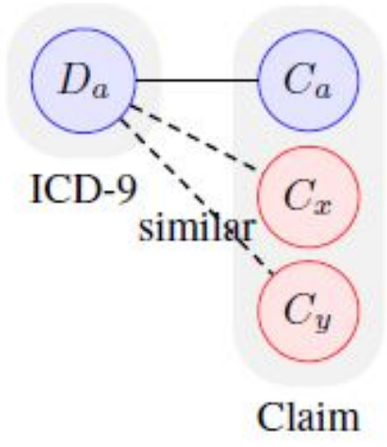
Shared providers



Collaborative filtering



Self-referral



Similarity search

SQL vs Cypher (Neo4J)

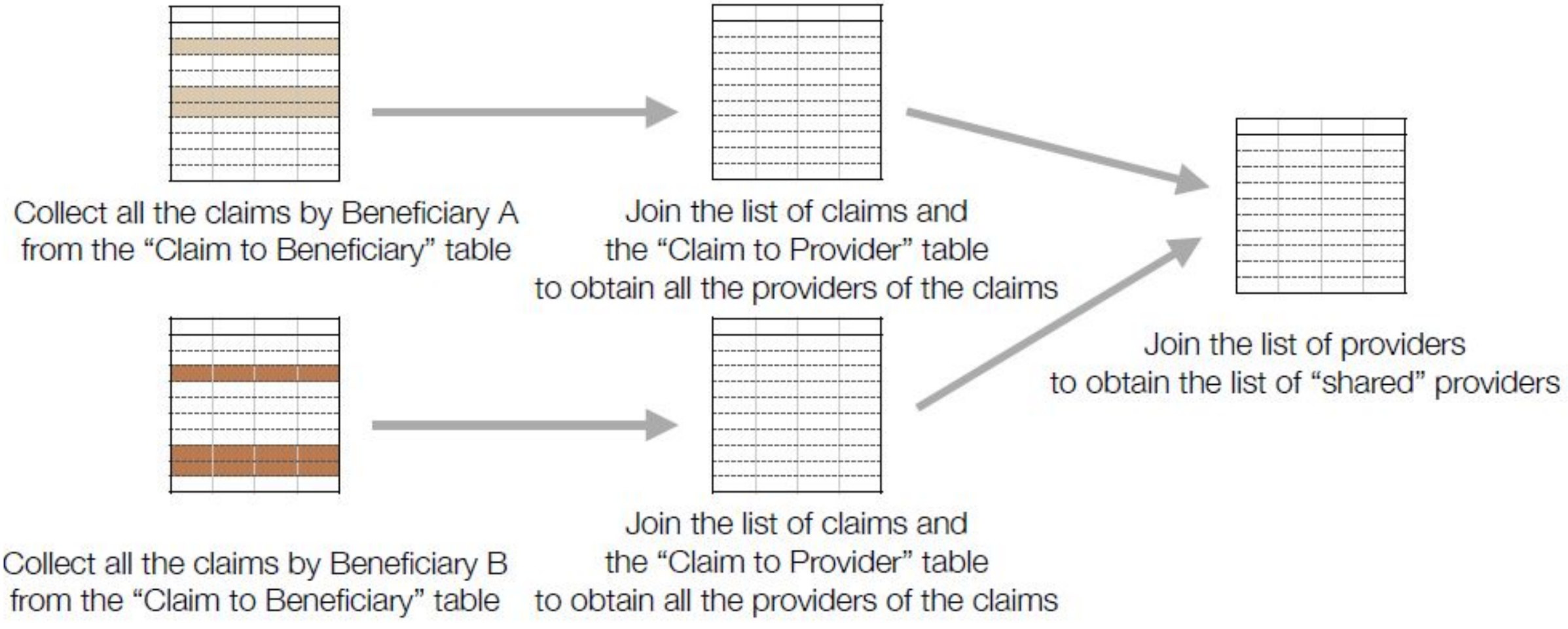
- To find a shared provider:
  - SQL

```
SELECT tableA.provID FROM (
  SELECT provID FROM C2P INNER JOIN (
    SELECT claimID FROM C2B WHERE
beneID=25869
  ) AS tmp ON tmp.claimID=C2P.claimID
 ) AS tableA INNER JOIN (
   SELECT provID FROM C2P INNER JOIN (
     SELECT claimID FROM C2B WHERE
 beneID=751751
   ) AS tmp ON tmp.claimID=C2P.claimID
 ) AS tableB ON tableA.provID=tableB.provID;
```

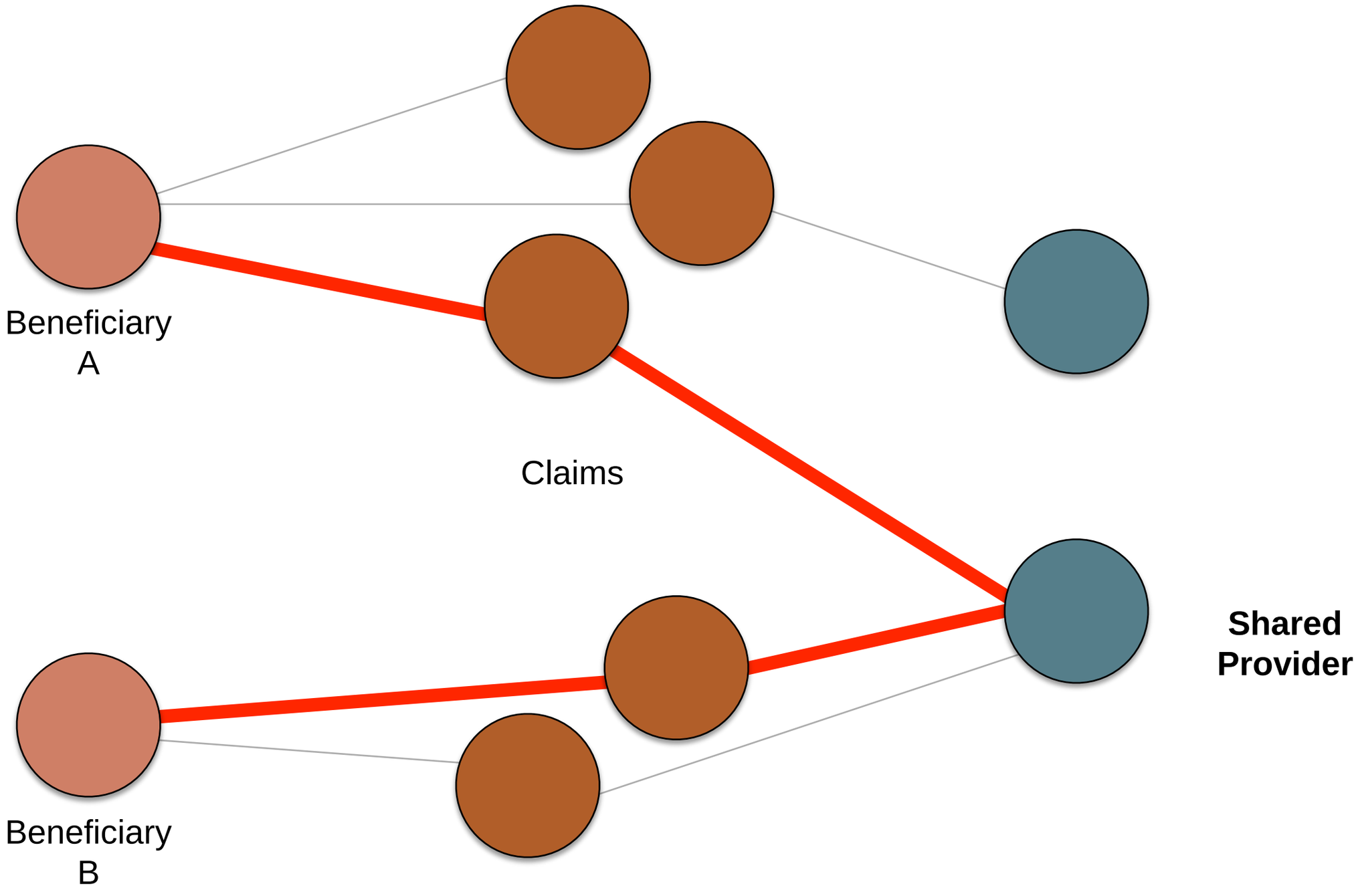
  - Cypher

```
START beneA=node(25869),
beneB=node(751751)
MATCH beneA-->()<--prov-->()<--beneB
RETURN prov;
```

To find a shared provider (SQL ver.)



To find a shared provider (Cypher ver.)



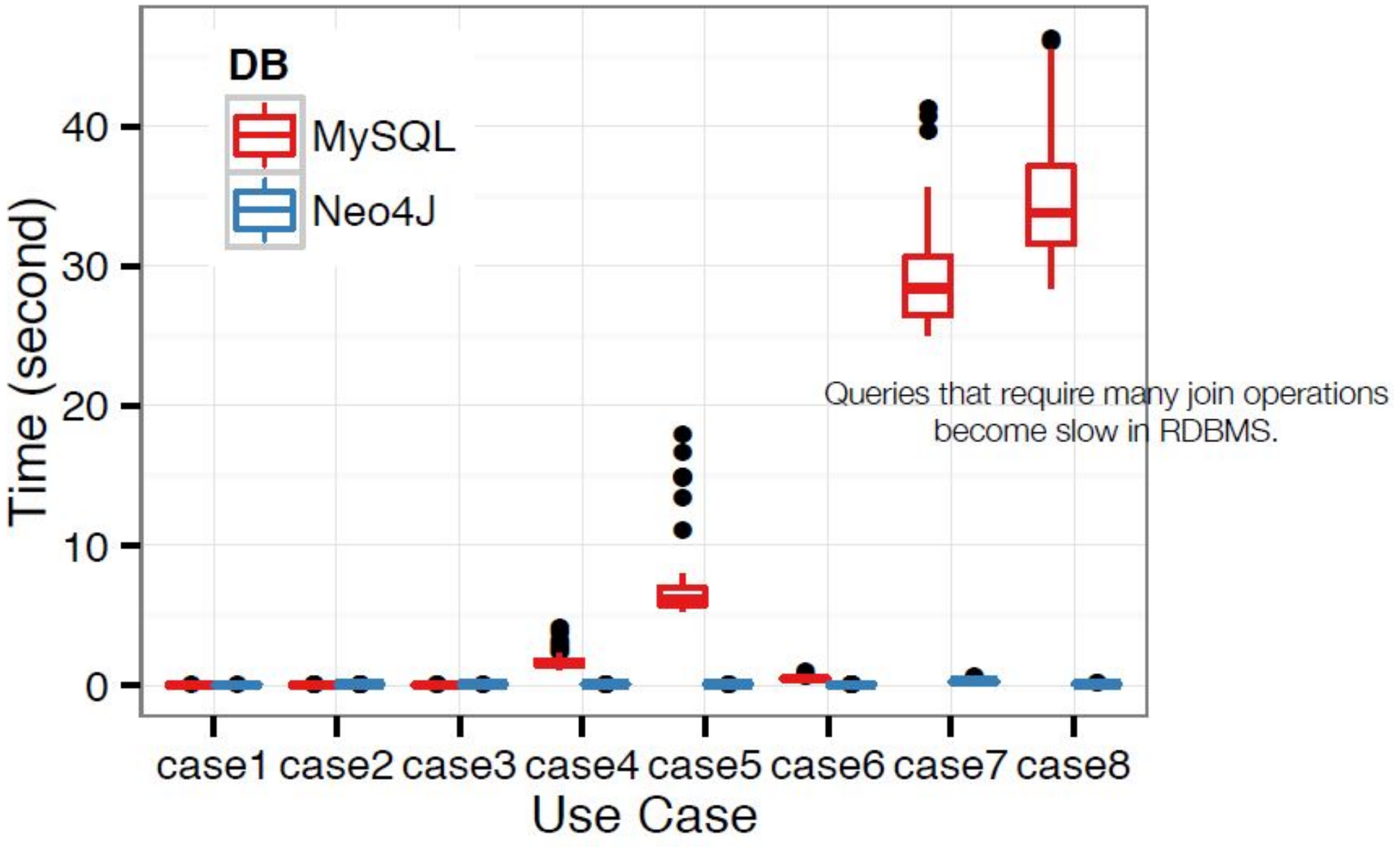
Execution on Use-Cases

- Case 1: Shared Provider
- Case 2: Loosely Specified Relationship
  - Shared entity through actual or referrals
- Case 3: Shared Diagnosis Code (similar to Case 1)
- Case 4: Any Link between Entities
- Case 5: Shared Beneficiary
- Case 6: Self Referral
- Case 7: Similar Record (based on diagnosis codes)
- Case 8: Collaborative Filtering (recommendation based on prior providers)

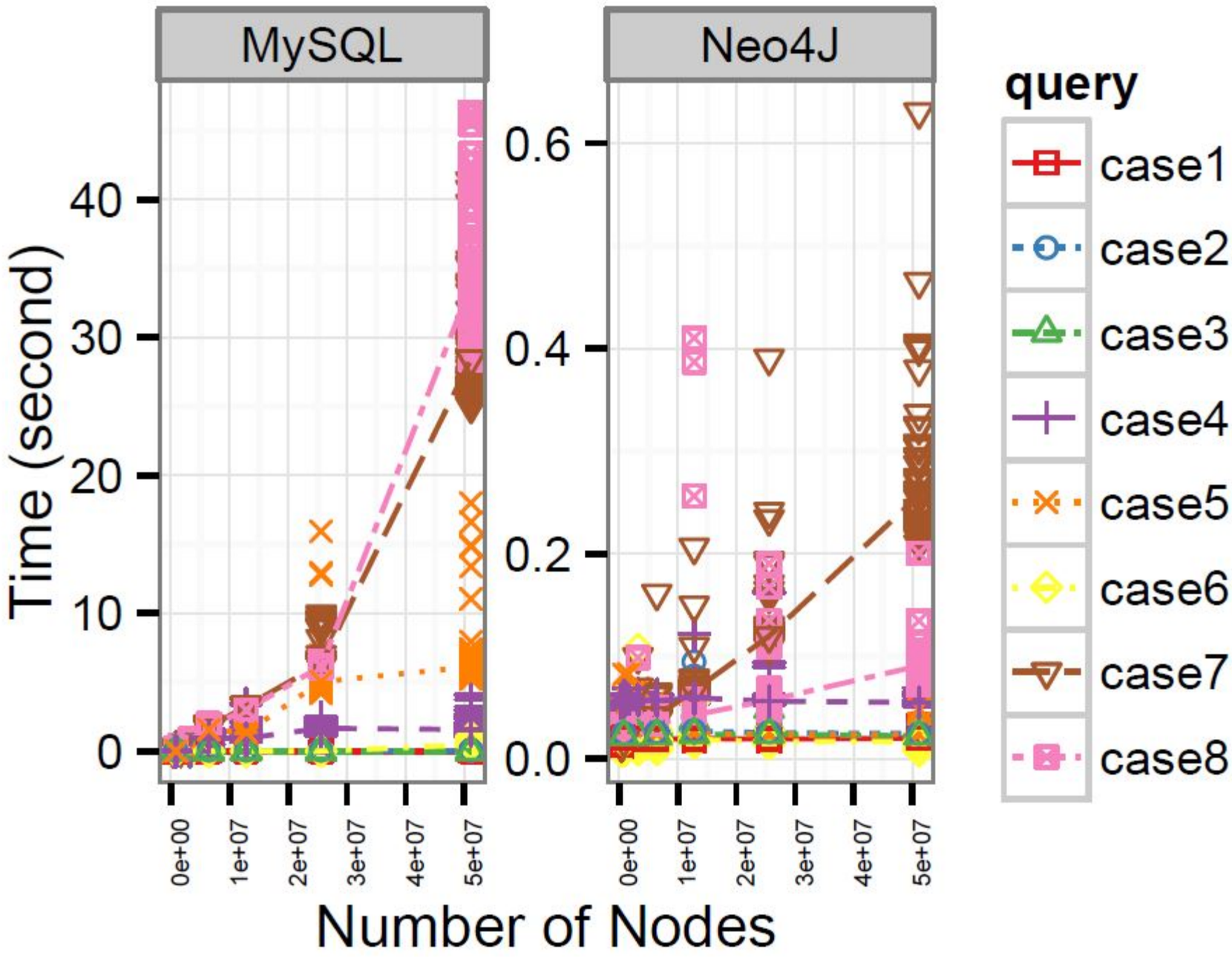
Test Data

Dataset	#. Bene	#. Prov	#. Claims	#. Nodes	#. Rel.
1	8K	800	400K	425K	2M
2	16K	1.6K	800K	824K	4M
3	31K	3.1K	1.5M	1.62M	8M
4	63K	6.3K	3.1M	3.21M	16M
5	125K	12.5K	6M	7M	33M
6	250K	25K	12M	13M	65M
7	500K	50K	25M	26M	129M
8	1M	100K	50M	51M	257M

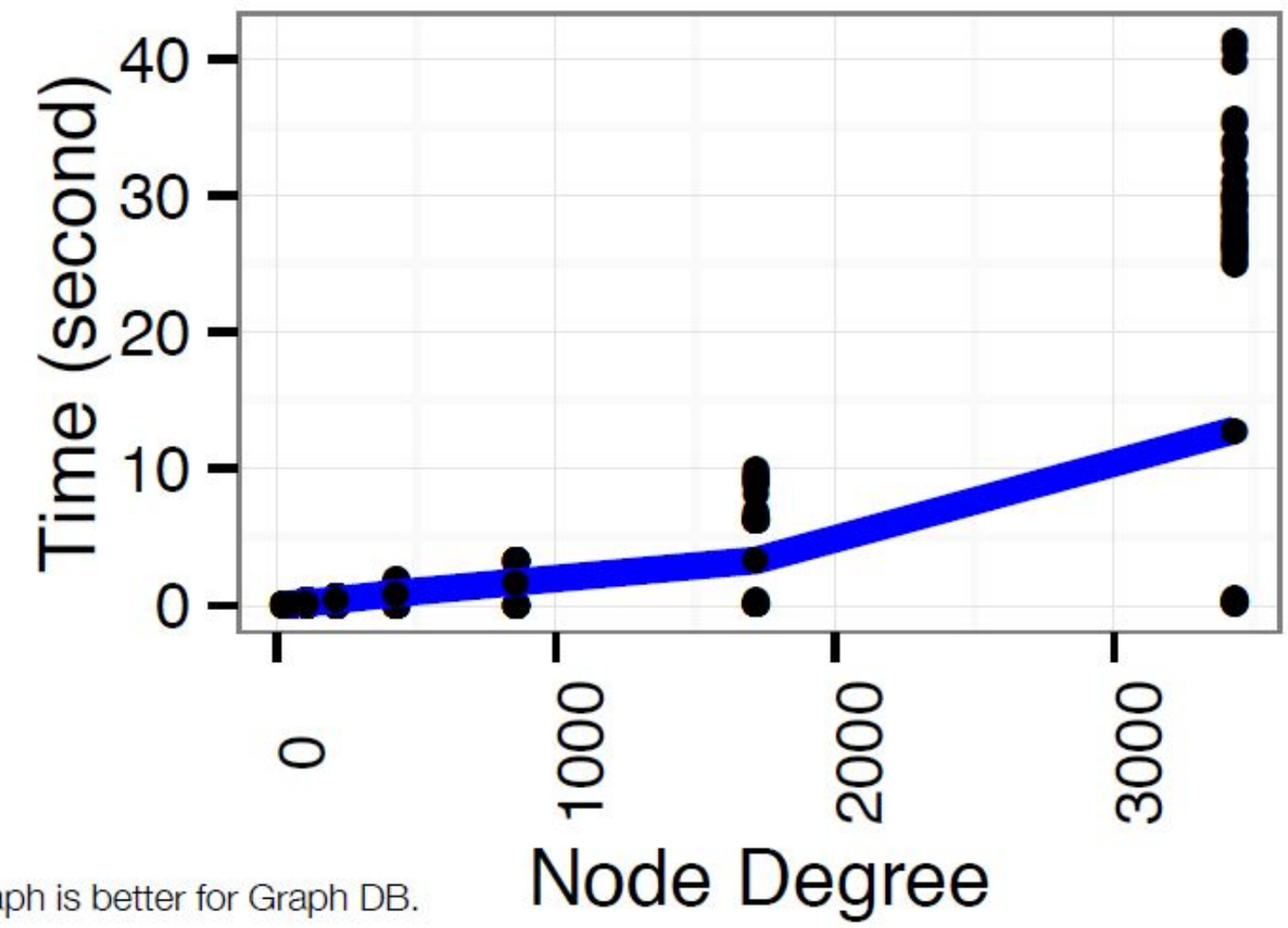
Performance Comparison



Increasing the Scale

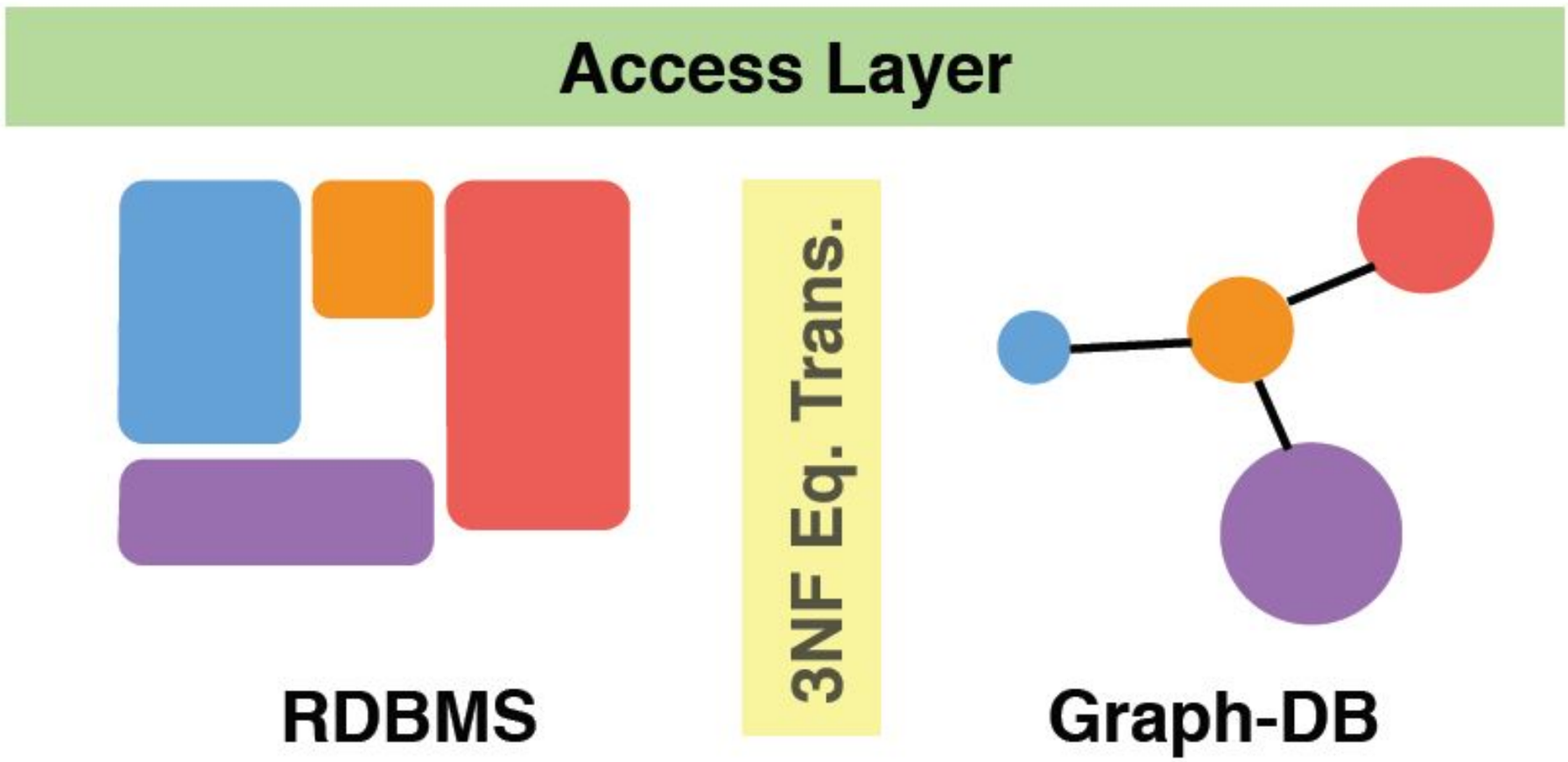


Increasing the Degree in the Graph



A sparse graph is better for Graph DB.

Transition Path with a Shared Layer



Implications

- Graph database as de-normalized tables
- Data integration using graph database
- Challenges for further scalability

Thank you!

Questions?

Contact:
shankarm@ornl.gov
yubin.park@utexas.edu

This work is sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory. Oak Ridge National Laboratory is managed by UT-Battelle, LLC for the U.S. D.O.E. under the contract no. DE-AC05-00OR22725