Neo4j: A Graph Database

Presenter: Jinliang Wei

CMU CSD

November 13, 2013

Why Graph Database?

Relational and NoSQL databases lack support for relationships.

- Relational databases: expensive joins.
- NoSQL databases: embed related keys in value.

Graph Database

- Designed for handling relationships.
- Process query by graph traversal from individual elements. Graph traversal is efficient.
- Avoid expensive joins.
- Query processing time is propotional to how much of the graph that query explores instead of the size of data stored.
- Example query: who are friends of James Bond's friends?

Neo4j

- Most popular graph database.
- Used by: Adobe, Cisco, Glassdoor, Huawei, HP...

Neo4j Data Model

- Graph model: property graph.
- Nodes.
- Relationships (edges) connect nodes.
- Both have properties.

Features of Neo4j

- ACID transactions.
- Distributed.
- Supports API in many languages: Java, Python, Ruby, JS...

A Neo4j Example - Cypher Query

```
MATCH (john {name: 'John'})-[:friend]->()-[:friend]->(fof)
RETURN john, fof
```

Performance: Neo4j vs. RDBMS

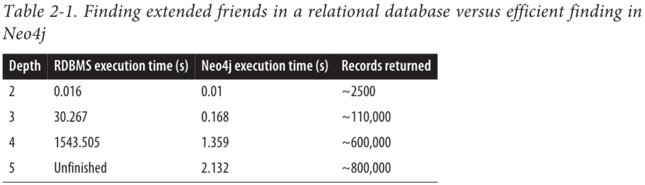

Table 2-1. Finding extended friends in a relational database versus efficient finding in Neo4j

Depth	RDBMS execution time (s)	Neo4j execution time (s)	Records returned
2	0.016	0.01	~2500
3	30.267	0.168	~110,000
4	1543.505	1.359	~600,000
5	Unfinished	2.132	~800,000

[1]

[1] reference: Graph Databases, p20