Neo4j Graph DB

{A step ahead towards non-RDBMS}

Presented By:
Mangal Dev

Agenda

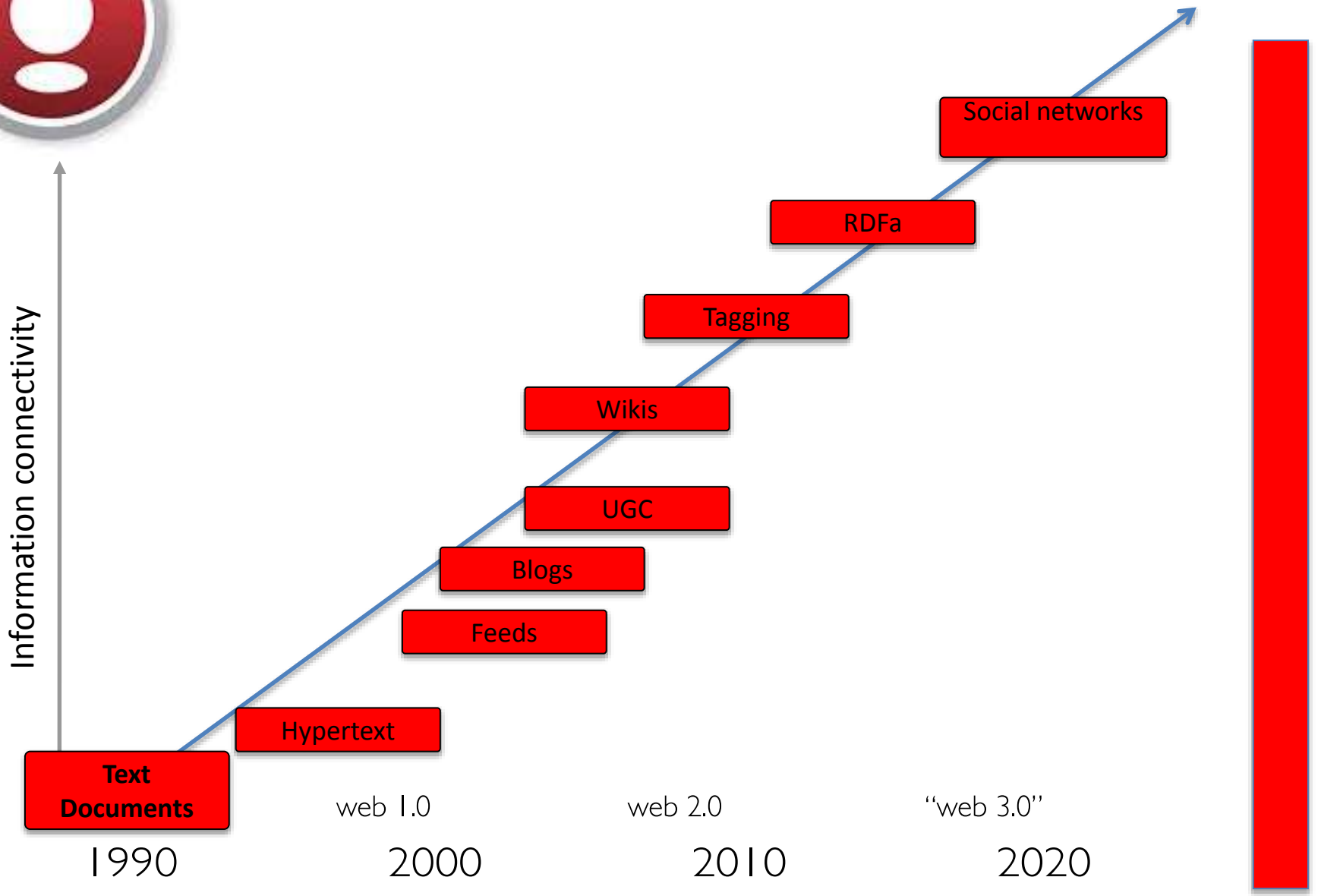
- N0SQL
- Graph Database
- Neo4J
- Samples

Brew install neo4j

Not Only SQL

Why ?

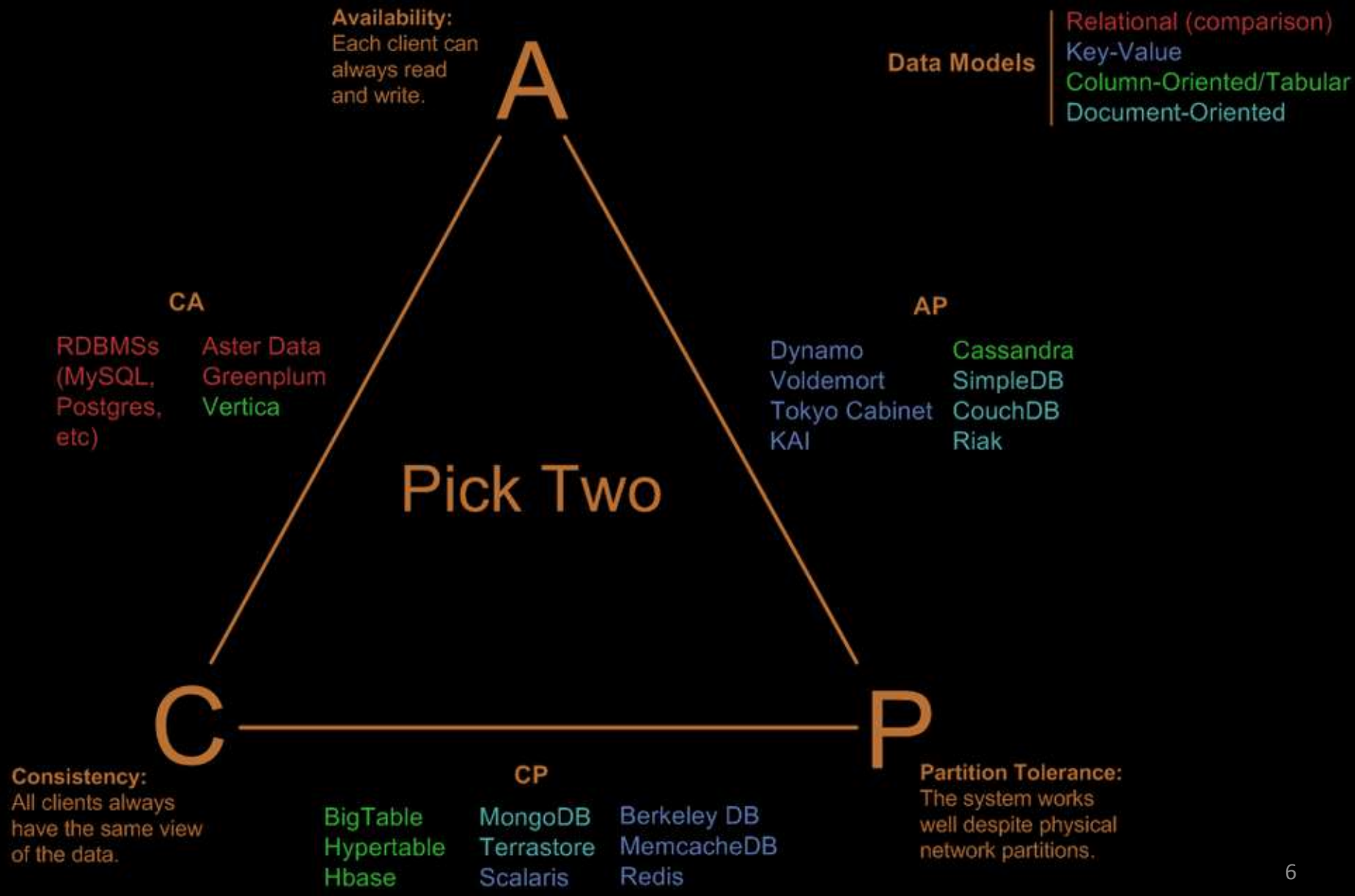
Interdependency and complexity of data.



NOSQL Categories

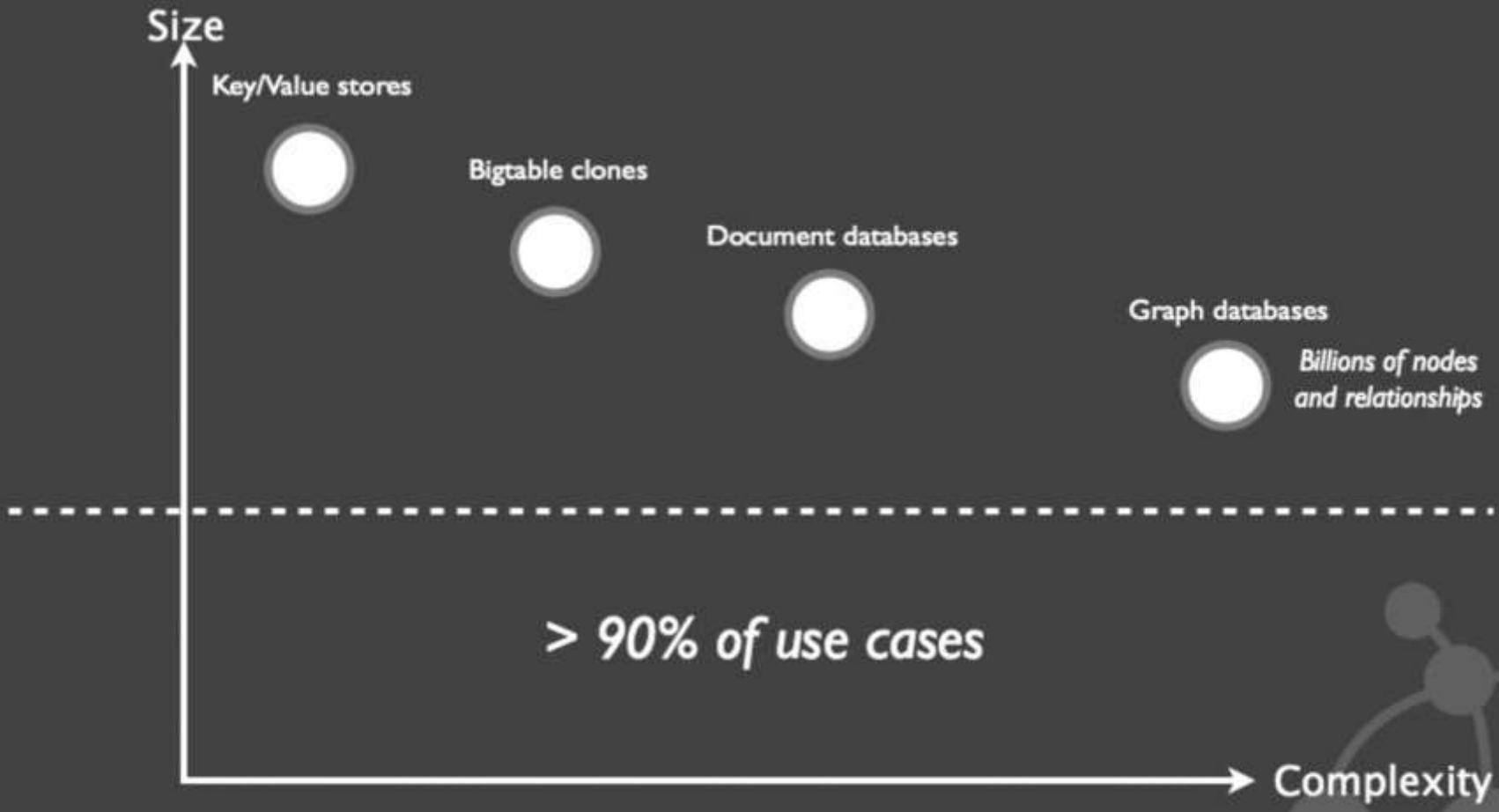
- **Key-Values Stores**
  - Dynamo
  - Riak
- **Column Family Stores**
  - Big Table -> Google
  - Casandra -> Facebook
  - HBase -> Apache
- **Document Based DB**
  - CouchDB
  - MongoDB
- **Graph Databases.**
  - AllegroDB
  - FlockDB -> Twitter
  - InfiniteGraph
  - Neo4j
  - Sones

Visual Guide to NoSQL Systems



neotechnology
value in relationships

Scaling to size vs. Scaling to complexity



Why GraphDB ?

- Flexibility
  No Normalization/De-normalization
  Relational databases deal poorly with relationships.

- Agility
  Schema free nature
  Foreign Keys turns into dangling refrences

- Reciprocal Queries

- Usecases
  SocialNetworks
  BioInformatics
  DataManagement
  NetworkManagement
  CloudManagement
  GeoData

What does ‘R’ stand for in RDBMS ?

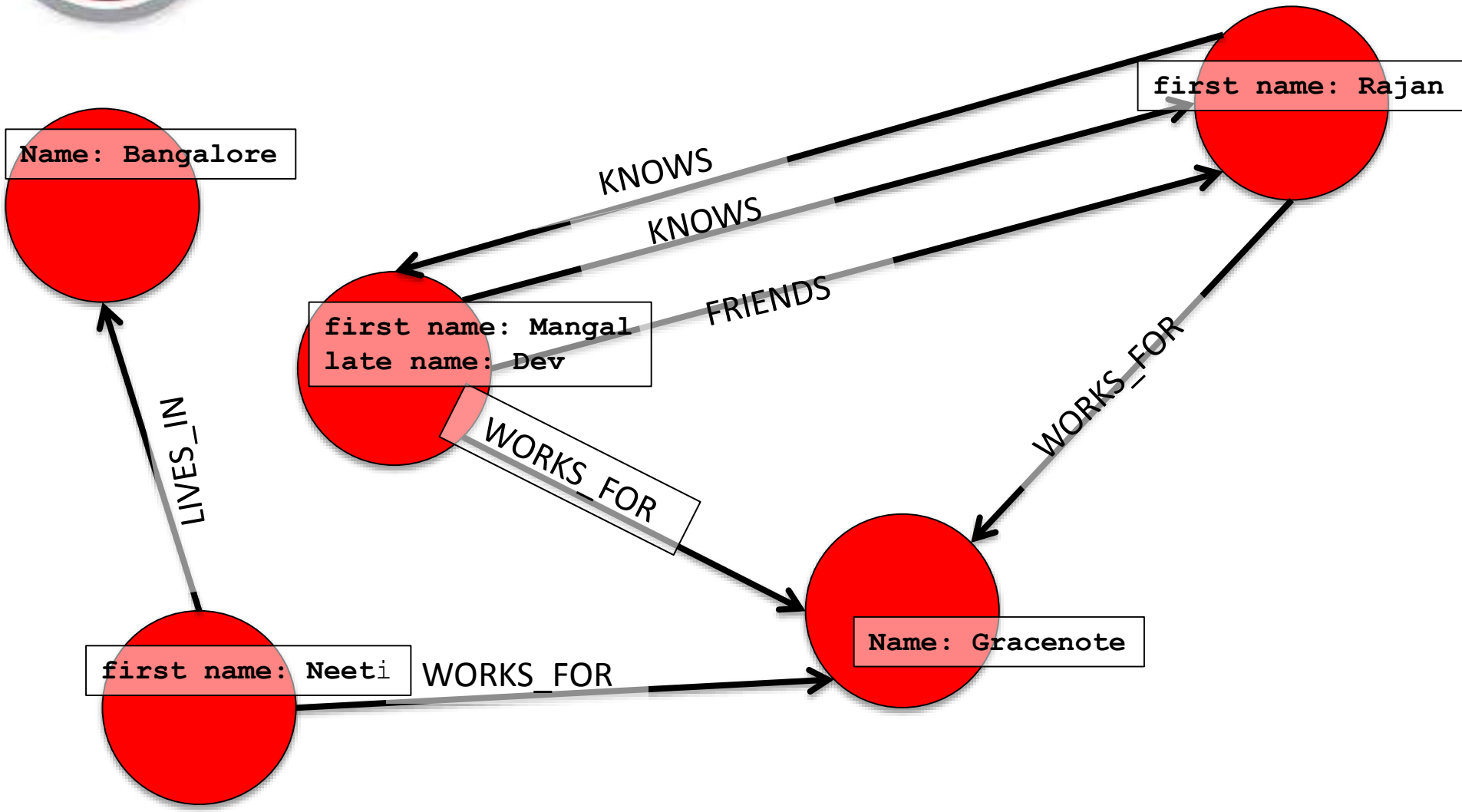
Graph Databases

sones

AllegroGraph

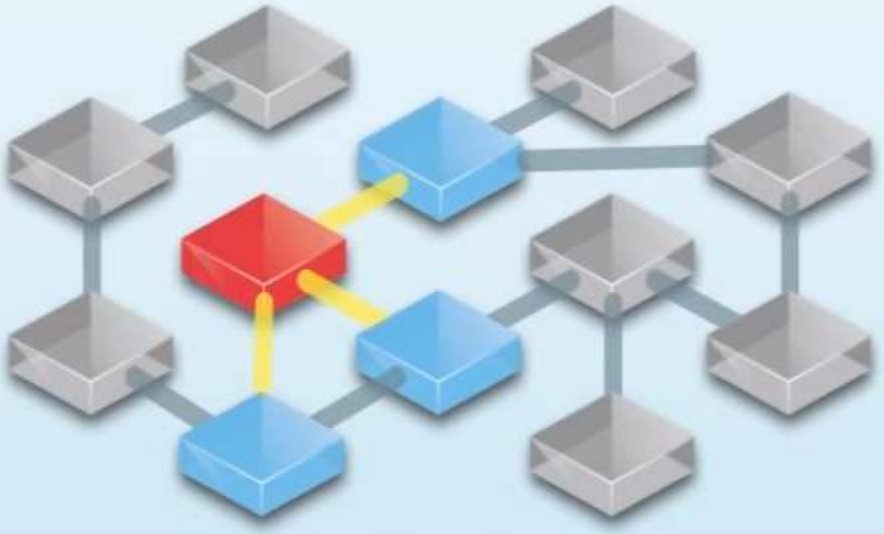
HYPERGRAPHDB

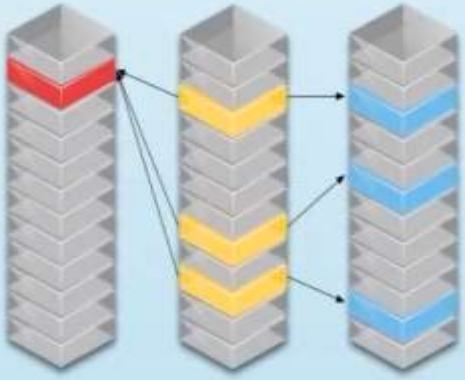
Property Graph Model

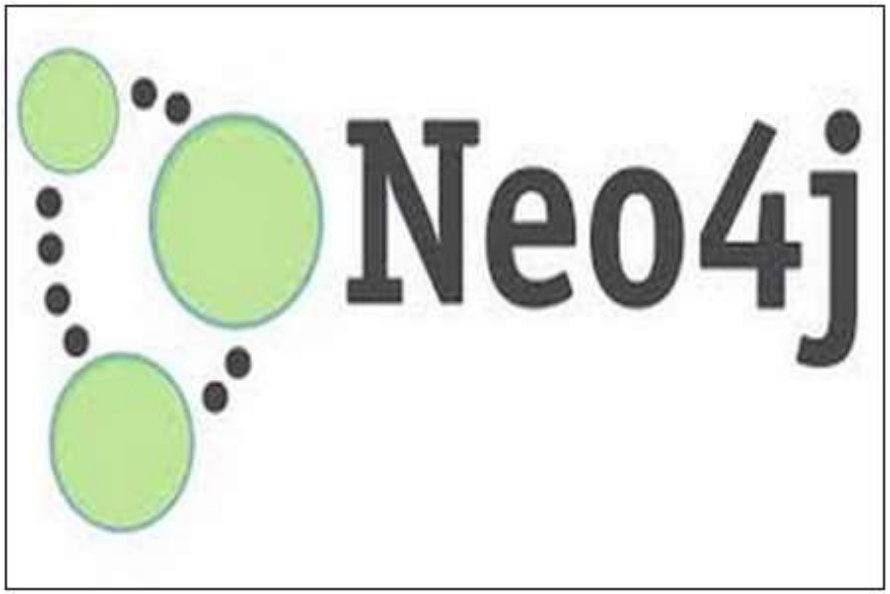


You know relational

now consider relationships...







Database full of linked nodes

Stores data as nodes and relationships

Some More Info..

- Widely Used
- Full ACID transactions
- Schema free, bottom-up data model design
- It’s a stable and in active developlment since mid 2000
- High Performance, can cover upto 2 billion nodes.
- Lot of Bindings

How do you query and work with this graph database?

- Run it as
  - Embedded
  - Standalone
- Query it with
  - Cyhper from Neo4j Browser
  - Cypher from Java
  - Rest API
  - Rest API using Cypher query
  - Programatically
  - Gremlin

Bindings

REST://

Cypher Query

- MATCH
  - Matches the graph pattern in the real graph.
- WHERE
  - Filters using predicates or anchors pattern elements.
- RETURN
  - Returns and projects result data, also handles aggregation.
- ORDER BY
  - Sorts the query result.
- SKIP/LIMIT
  - Paginates the query result.

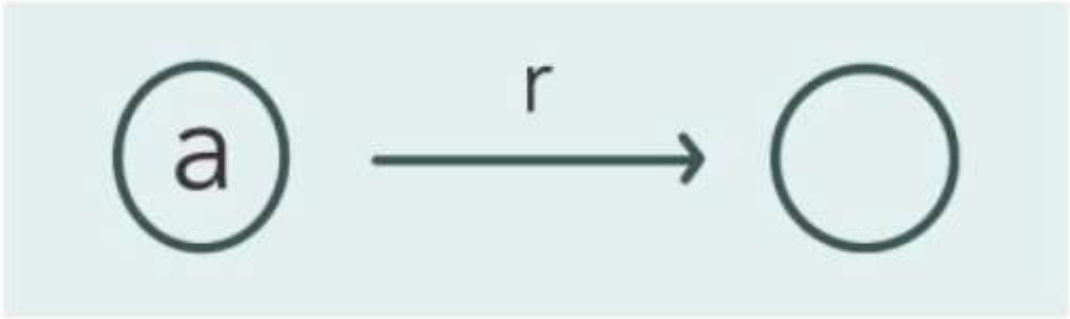
Go to Console…

- Neo4j start
- Go to url :
  - http://localhost:7474/browser/

Cypher Basics


```
MATCH  (a)-->(b)
RETURN a, b;
```

Naming a Relationship




```
MATCH (a)-[r]->( )
RETURN a.name, type (r);
```

Matching by Relationship

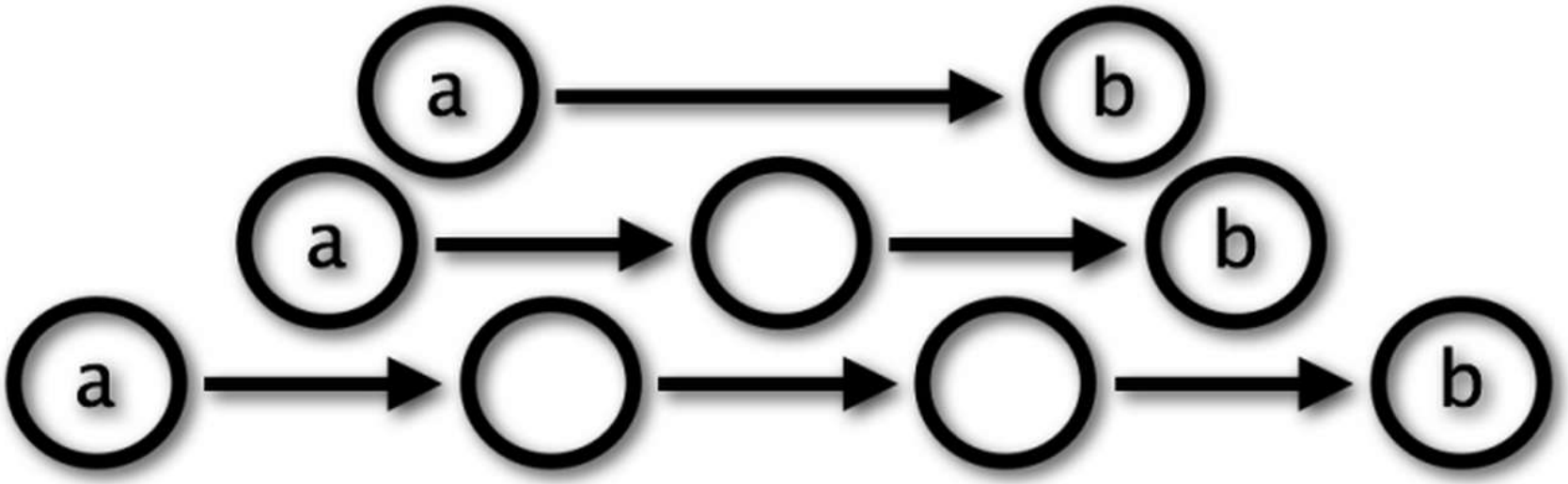

```
MATCH(a)-[:ACTED_IN]->(m)
RETURN a.name, m.title;
```

Match Node Label

```
MATCH (tom:Person)
WHERE tom.name="Tom Hanks"
RETURN tom;
```

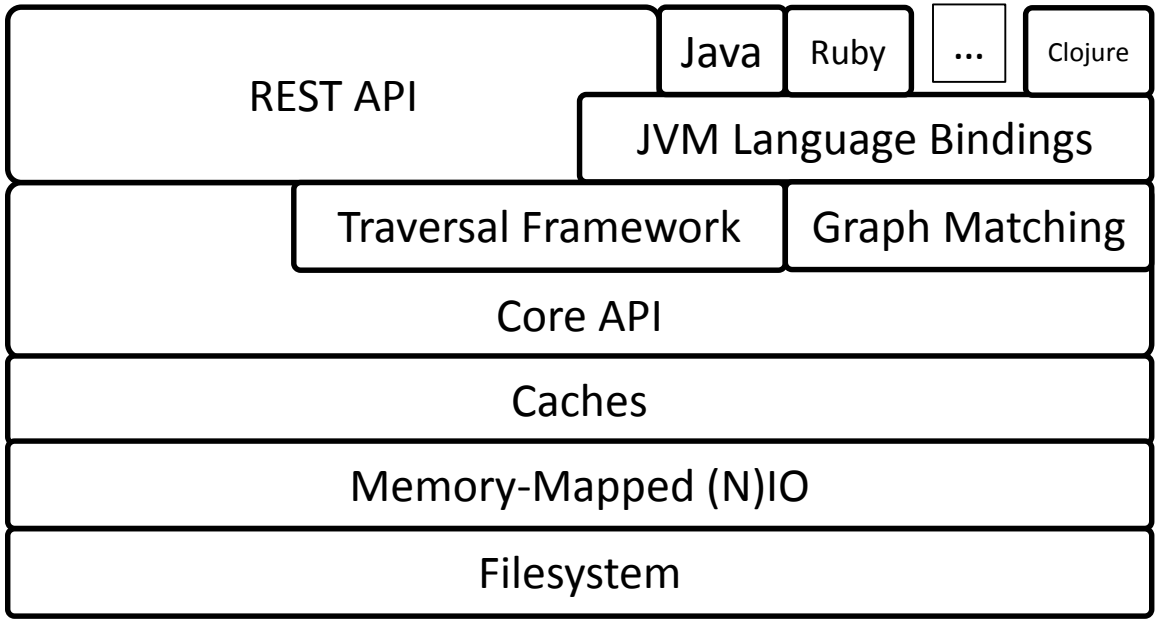
`:Person` – matches only nodes labeled as Person

Variable length paths




```
(a)-[*1..3]->(b)
```

Neo4j Logical Architecture



Graph Algorithms

- algorithm can have one of these values:
  - shortestPath
  - allSimplePaths
  - allPaths
  - dijkstra (optionally with cost_property and default_cost parameters)

Graph Algorithms

- shortestPath()

```
MATCH p =
ShortestPath
    ( (keanu:Person)-[:KNOWS*]-(kevin:Person) )
 WHERE
        keanu.name="Keanu Reeves"
                and
        kevin.name = "Kevin Bacon"
  RETURN
        length(p)
```

Indexing a Graph?

- Graphs are their own indexes!
- But sometimes we want short-cuts to well-known nodes
- Can do this in our own code
  - Just keep a reference to any interesting nodes

    e.g.

    CREATE INDEX ON :MOVIE(title);

    Don’t index every node!

Inbuilt Lucene Support

- Default index implementation for Neo4j
- Each index supports nodes *or* relationships
- Supports exact and regex-based matching
  - Query('index_name','pattern')
- Supports scoring
  - Number of hits in the index for a given item
  - Great for recommendations!

Scalability

- The HA component supports master-slave replication
- But Scaling graphs are **HARD.**
- Acc to Emil -> Superb performance till some billions of nodes

PEFORMANCE ANALYSIS

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

Pros and Cons

- Strengths
  - Powerful data model
  - Fast
    - For connected data, can be many orders of magnitude faster than RDBMS
- Weaknesses:
  - Sharding
    - Though they *can* scale reasonably well
    - And for some domains you can shard too!

ENOUGH THEORY!!!

Sample Exercises

- Find all Persons who killed other in Mahabharat.
- Find movies in which actor has acted along with directed it.
- the five actors who have acted in the most movies
- Shortest path between Kunti and Sehdev
- Find persons got killed by Kunti's sons

References

- http://www.neo4j.org/learn/online_course
- http://www.infoq.com/articles/graph-nosql-neo4j
- http://www.manning.com/partner/Neo4J_meap_ch01.pdf