# Graph Databases

- Relationships become first class objects
- c.f. relational- relationships are implicit in foreign keys, cross-reference tables, joins
- Difficult to traverse arbitrary graph structures

## Graph Model: Neo4j

### Nodes

- Labels (l or move)
- Properties (key:value)

### Relationships (directed edge between 2 nodes)

- Type (1 exactly)
- Properties (key:value)

### Applications

- Road networks/transport/geo-spatial
- Recommender systems (e.g. Netflix)
- Organizational structures
- Data networks/data centers

### Querying (Neo4j's Cypher

- Uses descriptive pattern matching
- Patterns: 'ASCII art' depictions of graphs structures

# Big Data

- Volume
- Velocity
- Variety- data trustworthiness
  - No validation
  - Statistical methods necessary
- Geared toward analytics and

# Map Reduce and Hadoop

- 2003: Google File System
- 2004: Map Reduce
- 2005: Hadoop
  - Filesystem
  - Map reduce platform → Yahoo! → 2011 open source (Apache)

## Map Reduce Concepts

- Paradigm:
  - Specify some operations to perform on collections of data items distributed over multiple nodes (map)
  - Collate answers together for final answer (reduce)
- Hadoop:
  - Provides a platform for automating Map Reduce jobs
  - Data distributionn
  - Parallelization
  - Fault tolerance/recovery
  - Communication
  - Load balancing
- Map: takes in a key:value pair → outputs multiple intermediate key:value pairs
- Reduce: take a key and a collection of all values associated wit key → output key:value pair
- E.g.: word count

## HDFS: Hadoop Distributed Filesystem

- Posix-like filesystem
- Distributed with replication, fault tolerance
- Optimized for
  - High throughput
  - Tradeoff latency

- Data locality
- "Hadoop ecosystem"
  - Higher level intefaces built on HDFS/Map Reduce
  - SQL database - Hive
  - Machine Learning - Apache Mahout
  - Business Intelligence - Apache Drill

## Apache Spark

- Same goals as map reduce
- More flexible/powerful framework than Hadoop