

Session 10

Graph Databases

Big Data Analytics Technology, MSc in Data Science,
Coventry University UK

Miyuru Dayarathna

Presentation Outline

- Introduction
- Native Graph Storage
- Graph Query Languages
- Performance and Scalability
- Conclusion



Property Graph Data Model

- It contains nodes and relationships
- Nodes contain properties (key-value pairs)
- Relationships are named and directed, and always have a start and end node
- Relationships can also contain properties

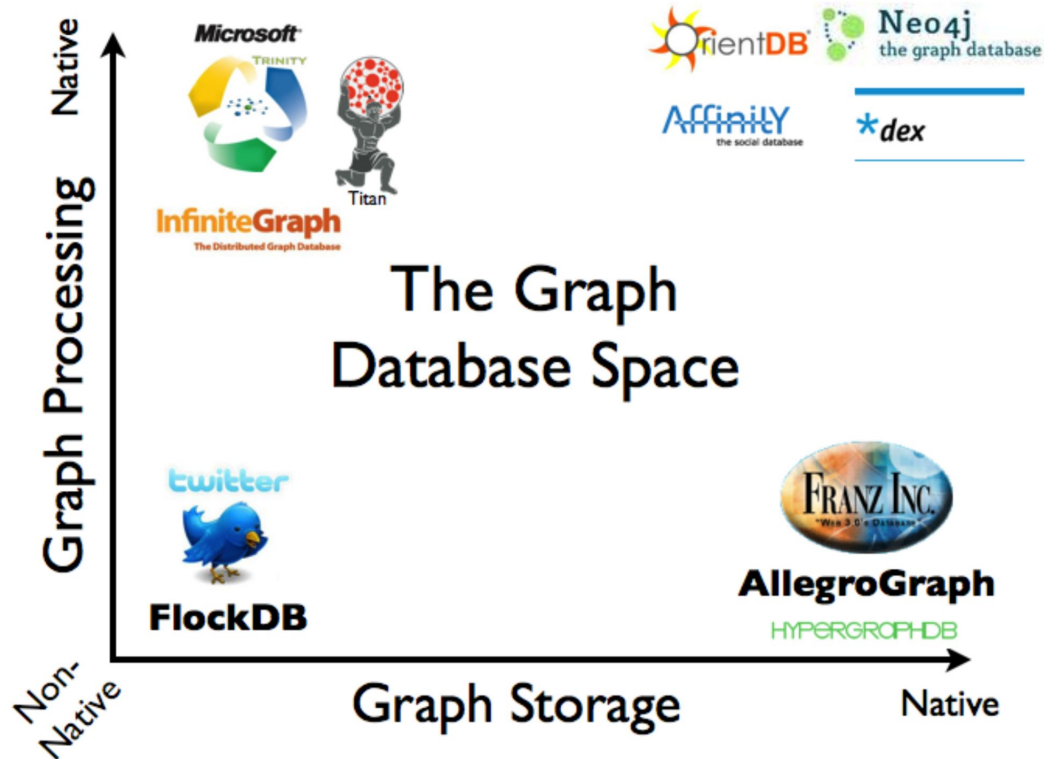
Graph Databases

- A graph database management system (henceforth, a graph database) is an online database management system with Create, Read, Update, and Delete (CRUD) methods that expose a graph data model.
- Graph databases are generally built for use with transactional (OLTP) systems.
- Accordingly, they are normally optimized for transactional performance, and engineered with transactional integrity and operational availability in mind.

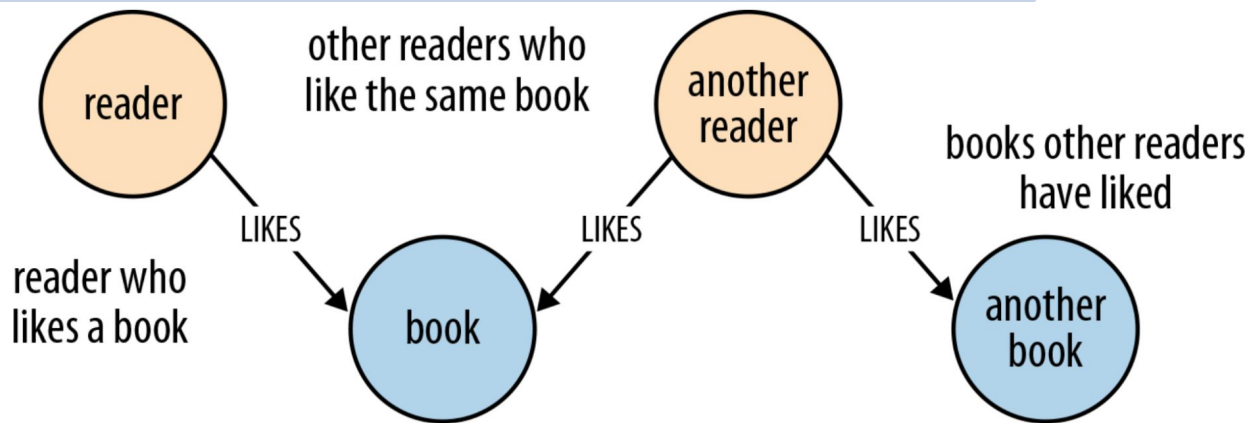
Two key properties of Graph Databases

- The underlying storage
 - ▷ Some graph databases use native graph storage that is optimized and designed for storing and managing graphs. Not all graph database technologies use native graph
 - ▷ storage, however. Some serialize the graph data into a relational database, an object oriented database, or some other general-purpose data store.
- The processing engine
 - ▷ the significant performance advantages of index-free adjacency, and therefore use the term native graph processing to describe graph databases that leverage index-free adjacency

Overview of Graph Database Space



Graph Query Languages



START reader=**node**:users(name={readerName})

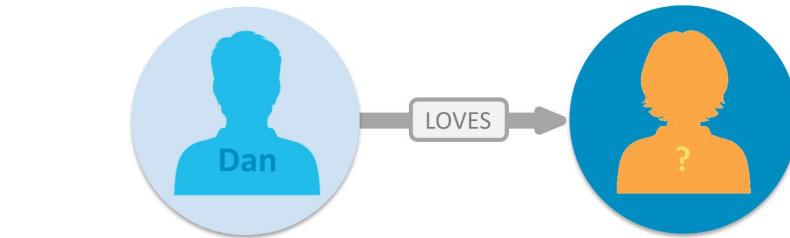
book=**node**:books(isbn={bookISBN})

MATCH reader-[:LIKES]->book<-[:LIKES]-other_readers-[:LIKES]->books

RETURN books.title

Cypher Query Language

- Cypher is Neo4j's graph query language that lets you retrieve data from the graph.
- It is like SQL for Graphs



MATCH (:Person { name:"Dan" }) -[:LOVES]-> (whom) **RETURN** whom

LABEL

PROPERTY

VARIABLE

Distributed Graph Database Servers

- Requirement
- Implementations
 - ▷ JasmineGraph
 - ▷ TigerGraph
 - ▷ Amazon Neptune
 - ▷ Neo4j Distributed

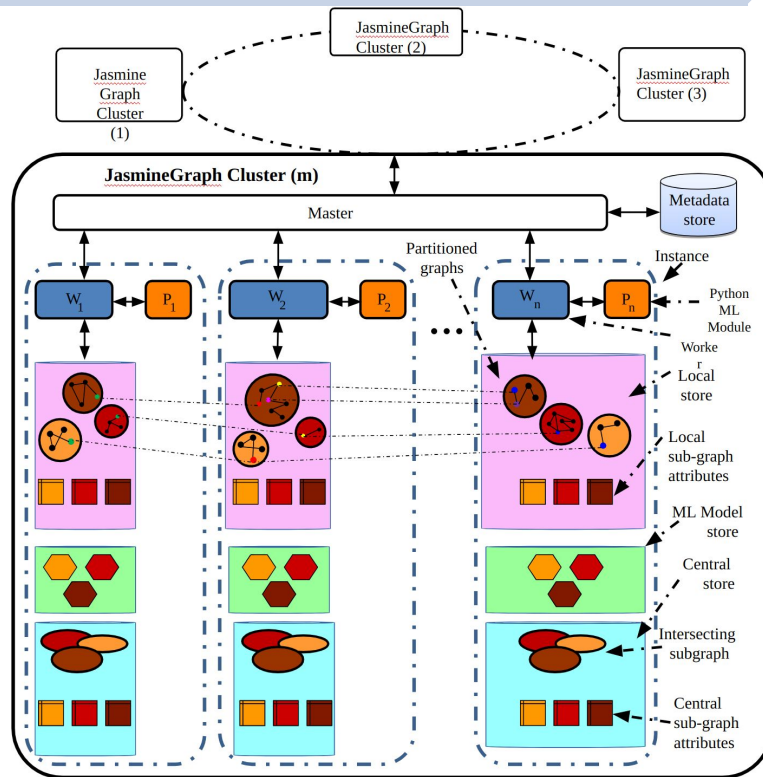
JasmineGraph

- An open source distributed graph database server
<https://github.com/miyurud/jasminegraph>

The screenshot shows the GitHub repository for JasmineGraph, a distributed graph database server. The repository is public and has 16 branches and 1 tag. The main branch is master. The repository has 1,403 commits, 21 forks, and 16 stars. The repository is licensed under Apache-2.0. The repository is categorized under graph, databases, high-performance-computing, graph-theory, graph-database, graphdb, data-management, graphdatabase, and graph-algorithms. The repository has a README, 1 tag, and 1 release. The repository has 6 contributors and 362 deployments.

File	Description	Last Commit
github	Upgrade action versions	3 weeks ago
cmake_modules	Configure code coverage	2 months ago
conf	Include DDL refactorization of streamingdb	last month
docs	Update index.html	5 years ago
k8s	Add a wrapper script for start k8s	3 weeks ago
python	Add machine learning related folder structure	4 years ago
samples/ml	Add machine learning related folder structure	4 years ago
src	Merge branch 'master' into k8s	2 weeks ago
src_python	Enforce quote checks	2 months ago
tests	Fix role problem in reading node list	3 weeks ago
dockerignore	Copy metadata in to the unit test docker container	last month
.gitignore	Ignore database files	3 weeks ago
.pylintrc	Enforce quote checks	2 months ago
CMakeLists.txt	Apply suggestions from code review	3 weeks ago
Dockerfile	Use python3 symlink	last month
LICENSE	Initial commit	6 years ago
README.md	Add a wrapper script for start k8s	3 weeks ago
backup.sh	Fix issues	last month
build.sh	Fix issues	last month

JasmineGraph System Architecture



Graph Triangle Counting - Practical Session

- Download the following graph and upload to JasmineGraph
- Count the number of triangles using *trian* command of JasmineGraph and verify the results obtained with the reported triangle count in the below link

<https://snap.stanford.edu/data/email-Enron.html>

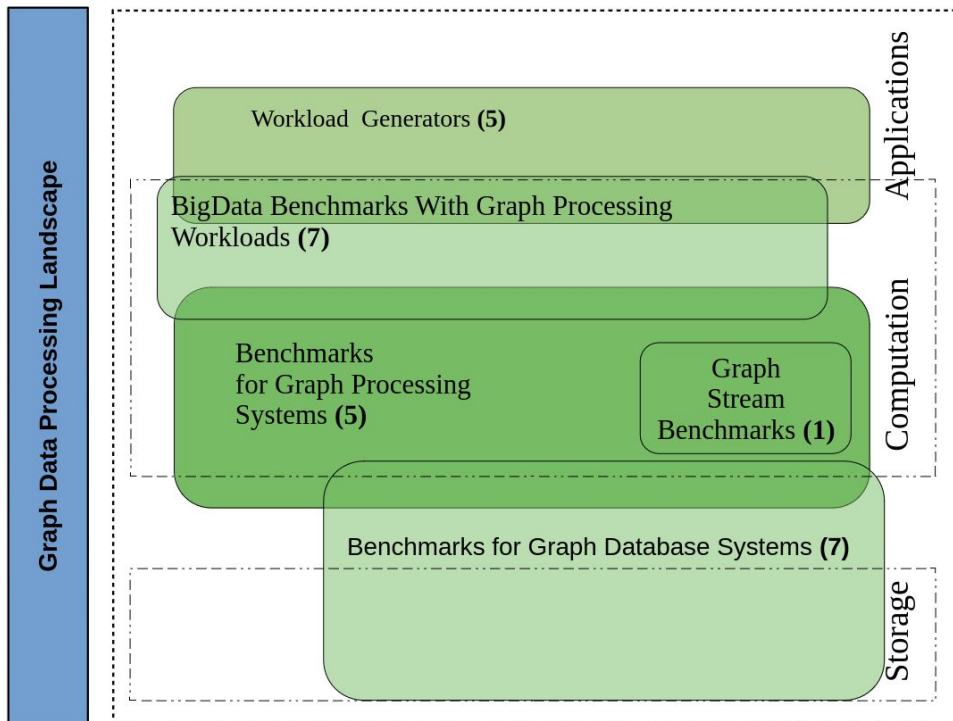


Practical Session

Graph Processing System Performance Measurement

- Graph 500
- LDBC (Linked Data Benchmark Council)
- Graphalytics

Summary view of graph benchmarking landscape



Graph Data Benchmarking Landscape

<https://arxiv.org/pdf/2005.12873.pdf>

Thank you!