

Benchmarking of Graph Databases - Suitability for the Industrial Environment

Master's Thesis
by

Christian Navolskyi

Chair of Pervasive Computing Systems/TECO
Institute of Telematics
Department of Informatics

First Reviewer:	Prof. Dr. Michael Beigl
Second Reviewer:	M.Sc. Andrei Miclaus
Supervisor:	

Project Period: 01/01/2018 – 30.04.2018

Ich versichere wahrheitsgemäß, die Arbeit selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde.

Karlsruhe, den **TODO: date**

Zusammenfassung

TODO: Zusammenfassung (Deutsch)

Abstract

TODO: Zusammenfassung (Englisch)

Contents

1	Introduction	1
1.1	Problem Statement	1
1.1.1	Use Case - Industry 4.0	1
1.1.1.1	Inserting Data	1
1.1.1.2	Reading Data	1
1.2	Question	1
1.3	Methodology	1
1.4	Goal of this Thesis	1
1.5	Structure	1
2	Background & Related Work	3
2.1	Industrial Data	3
2.2	Graphs	3
2.3	Graph Databases	3
2.3.1	Triple Stores	3
2.3.1.1	Apache Jena	3
2.3.2	Document Stores	3
2.3.2.1	OrientDB	3
2.3.3	Graph Stores	3
2.3.3.1	Neo4j	3
2.3.3.2	Sparksee	3
2.4	Graph Database Benchmarks	3
2.4.1	LDBC: Graphalytics	3
2.4.2	XGDBench	3
2.4.3	YCSB	3
2.5	Related Work	3
2.5.1	Graph Database: Anna	3
2.5.2	TODO: Add more	3
3	Analysis	5
3.1	Data	5
3.1.1	Data Structure NOTE: Here or in Design?	5
3.1.2	Data Amount	5
3.2	Workloads	5
3.2.1	Inserting Data into the Database	5
3.2.2	Retrieving Data from the Database	5
3.3	Benchmark - YCSB	5
4	Design	7

4.1	Data Structure	7
4.2	Workloads	7
4.2.1	Inserting	7
4.2.2	Production Simulation	7
4.2.3	Reading under load	7
4.3	Extension of the Benchmark	7
4.3.1	Generating a Dataset	7
4.3.1.1	Storing the Dataset	7
4.3.1.2	Restoring the Dataset	7
4.3.2	Graph Workload	7
4.3.3	Bindings	7
4.3.3.1	Apache Jena	7
4.3.3.2	Neo4j	7
4.3.3.3	OrientDB	7
4.3.3.4	Sparksee	7
5	Implementation of your Project	9
5.1	Graph Workload	9
5.1.1	Parameters	9
5.2	Graph Data Generator	9
5.2.1	Parameters	9
5.2.2	Graph Data Creator	9
5.2.3	Graph Data Recreator	9
5.3	Graph Database Bindings	9
5.3.1	Apache Jena	9
5.3.2	Neo4j	9
5.3.3	OrientDB	9
5.3.4	Sparksee	9
6	Evaluation	11
6.1	Objective	12
6.2	Setup	12
6.2.1	Hardware	12
6.2.2	Software	12
6.3	Execution NOTE: Scripts to run all benchmarks successively	12
6.4	Maximum Load	12
6.4.1	Probing Node Count NOTE: Comparing indexed to not indexed	12
6.4.1.1	Results	12
6.4.1.2	Discussion	12
6.4.2	Probing Node Size NOTE: See change over increasing node size	12
6.4.2.1	Results	12
6.4.2.2	Discussion	12
6.5	Throughput	12
6.5.1	Difference of Presence of Edges	12
6.5.1.1	Results	12
6.5.1.2	Discussion	12
6.5.2	Product Complexity NOTE: More child nodes.	12
6.5.2.1	Results	12
6.5.2.2	Discussion	12

6.5.3	Production Suitability	NOTE: Testing production like workload	12
6.5.3.1	Results		12
6.5.3.2	Discussion		12
6.6	Responsiveness		12
6.6.1	Reading under load		12
6.6.1.1	Results		12
6.6.1.2	Discussion		12
6.6.2	Scanning under load		12
6.6.2.1	Results		12
6.6.2.2	Discussion		12
7	Conclusion and Future Work		13
7.1	Conclusion		13
7.1.1	Suitability		13
7.1.2	General Performance of Databases		13
7.2	Future Work		13
7.2.1	More Bindings		13
7.2.2	Concurrency		13
7.2.3	Other input methods	NOTE: I only used native Java APIs, to directly test the database.	13
7.2.4	Workload	TODO: what kind?	13
8	Summary		15

1. Introduction

1.1 Problem Statement

1.1.1 Use Case - Industry 4.0

1.1.1.1 Inserting Data

NOTE: How is that used by the industry.

1.1.1.2 Reading Data

NOTE: How is that used by the industry.

1.2 Question

1.3 Methodology

1.4 Goal of this Thesis

1.5 Structure

2. Background & Related Work

2.1 Industrial Data

2.2 Graphs

2.3 Graph Databases

2.3.1 Triple Stores

2.3.1.1 Apache Jena

2.3.2 Document Stores

2.3.2.1 OrientDB

2.3.3 Graph Stores

2.3.3.1 Neo4j

2.3.3.2 Sparksee

2.4 Graph Database Benchmarks

2.4.1 LDBC: Graphalytics

2.4.2 XGDBench

2.4.3 YCSB

2.5 Related Work

2.5.1 Graph Database: Anna

2.5.2 **TODO: Add more**

3. Analysis

3.1 Data

3.1.1 Data Structure **NOTE: Here or in Design?**

3.1.2 Data Amount

3.2 Workloads

3.2.1 Inserting Data into the Database

NOTE: What is the pattern of insertion

3.2.2 Retrieving Data from the Database

NOTE: What is the pattern of retrieving

3.3 Benchmark - YCSB

NOTE: Activity diagram in this section. What WAS the workflow.

4. Design

4.1 Data Structure

4.2 Workloads

4.2.1 Inserting

4.2.2 Production Simulation

4.2.3 Reading under load

4.3 Extension of the Benchmark

4.3.1 Generating a Dataset

4.3.1.1 Storing the Dataset

4.3.1.2 Restoring the Dataset

4.3.2 Graph Workload

NOTE: Graph Workload functionality NOTE: Activity diagram of the workflow with graph data.

4.3.3 Bindings

4.3.3.1 Apache Jena

4.3.3.2 Neo4j

4.3.3.3 OrientDB

4.3.3.4 Sparksee

5. Implementation of your Project

5.1 Graph Workload

5.1.1 Parameters

5.2 Graph Data Generator

5.2.1 Parameters

5.2.2 Graph Data Creator

5.2.3 Graph Data Recreator

5.3 Graph Database Bindings

5.3.1 Apache Jena

5.3.2 Neo4j

5.3.3 OrientDB

5.3.4 Sparksee

6. Evaluation

6.1 Objective

6.2 Setup

6.2.1 Hardware

6.2.2 Software

6.3 Execution **NOTE: Scripts to run all benchmarks successively**

6.4 Maximum Load

6.4.1 Probing Node Count **NOTE: Comparing indexed to not indexed**

6.4.1.1 Results

6.4.1.2 Discussion

6.4.2 Probing Node Size **NOTE: See change over increasing node size**

6.4.2.1 Results

6.4.2.2 Discussion

6.5 Throughput

6.5.1 Difference of Presence of Edges

6.5.1.1 Results

6.5.1.2 Discussion

6.5.2 Product Complexity **NOTE: More child nodes.**

6.5.2.1 Results

6.5.2.2 Discussion

6.5.3 Production Suitability **NOTE: Testing production like workload**

6.5.3.1 Results

6.5.3.2 Discussion

6.6 Responsiveness

6.6.1 Reading under load

6.6.1.1 Results

6.6.1.2 Discussion

6.6.2 Scanning under load

6.6.2.1 Results

6.6.2.2 Discussion

7. Conclusion and Future Work

7.1 Conclusion

7.1.1 Suitability

7.1.2 General Performance of Databases

7.2 Future Work

7.2.1 More Bindings

7.2.2 Concurrency

7.2.3 Other input methods **NOTE: I only used native Java APIs, to directly test the database.**

7.2.4 Workload **TODO: what kind?**

8. Summary

