

# 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:  
Second Reviewer:  
Supervisor:

Prof. Dr. Michael Beigl  
M.Sc. Andrei Miclaus

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 Problem Statement

With the growing digitalisation of the industry more data is available and can be used to improve production processes. The amount of data created depends on the individual use case, but still it needs to be stored to be useful. Since there are multiple databases available it can be difficult to choose the right one for an individual scenario.

### 1.1.1 Use Case - Industry 4.0

There are multiple analytic algorithms to run on data to extract certain features. In the industry those algorithms play an important role too, but in this thesis we are looking at different aspects of the industrial use case, mainly inserting data and reading data.

#### 1.1.1.1 Inserting Data

**NOTE:** [How is that used by the industry.](#)

To digitalise the production processes the data produced by every machine in the production line should be stored for future analysis. And to store that data it needs to be written into a database. Since most factories running 24 hours a day the machines are producing a lot of data during the day. That will be the base load for the underlying database, to store all that data from the production machines.

#### 1.1.1.2 Reading Data

**NOTE:** [How is that used by the industry.](#)

Besides using the stored data for analysis algorithms, simply reading data from the database is another common use case. An example would be to get the time at which a specific product was processed by a specific machine to check if all parameters were set correctly.

## 1.2 Question

This thesis should give an answer to the question, if graph databases are suitable for an industrial application. We concentrate on graph databases because of the structure of the production data. Graphs and the use of graph databases are further motivated in section ?? and section ?? respectively.

## 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 without 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 Workloads

TODO: what kind?





## 8. Summary

