

NoSQL Databases

Daniel Rutz and Leon Schürmann (Eds.)

March 23, 2019

Contents

| | | |
|----------|---|----------|
| 1 | GraphDB | 3 |
| | Thore Krüss, Lennart Purucker, Johanna Sommer | |
| 1.1 | Motivation/Introduction | 3 |
| 1.2 | Graph Database Theory | 5 |
| 1.2.1 | Description of data model and functionality | 5 |
| 1.2.2 | fields of application | 5 |
| 1.2.3 | CAP Theorem | 5 |
| 1.2.4 | GraphDB vs. RDBMS | 5 |
| 1.3 | Implementation with Neo4j | 5 |
| 1.3.1 | Use Case from the SQL world | 5 |
| 1.3.2 | Installation | 5 |
| 1.3.3 | modelling | 5 |
| 1.3.4 | usage, query language | 5 |
| 1.3.5 | short conclusion, summary | 5 |
| 1.4 | Reflection | 5 |
| 1.4.1 | alternative popular graphdbs | 5 |
| 1.4.2 | conclusion | 5 |

1 GraphDB

Thore Krüss, Lennart Purucker, Johanna Sommer

1.1 Motivation/Introduction

First paragraph to state the general context and make important points for the motivation and why its important.

But Graph Database research has been initiated already in the early 90s. During this time, a number of suggestions came up, trying to propose a semantic network to store data about the database. That was, because contemporary systems were failing to take into account the semantics of a database. The Locial Data Model [2] was proposed, where they tried to combine the advantages of relational, hierarchical and network approaches in that they modeled databases as directed graphs, in which leaves represent attributes and internal nodes as connections between the data.

Similar to that the Functional Data Model [?] was proposed with the same goal, focusing specifically on providing a conceptually natural database interface [1].

This this, most of the underlying theory of Graph Databases was created. It was most likely because of insufficient hardware support for big graphs that this research declined, only to be picked up again now, which is focused on actual practical systems because we have the hardware now but also on theoretical analysis of graph query languages [1].

Especially practical implementations of Graph Database Theory have gained traction, as real world problems are more often than not interrelated - hence graphs are extremely useful in understanding the wide diversity of real-world datasets.[?]

The emerging of social networks have naturally helped graph database implementations come up, with big players like Twitters FlockDB entering the field. In social network situations, a so-called social graph can model attributes of a person as well as relationships between people. While in traditional RDBMS the friend-of-a-friend-problem would be achieved via a join, in graph database technology this can be achieved over a traversal, which is far more cost inexpensive. [?].

Another field is recommender systems, where much work has been done in machine learning. But also in databases, this specific context poses challenges. But also here,

1 *GraphDB*

the graph model makes it easy to map item-to-item relationships in that they are similar to each other and can be recommended and also user-to-user, based on the correlations between the behaviour of two customers [?].

These application fields bring very specific workloads that need special functionality to process. There are two different kinds of workload: in social network transactions low-latency online graphs are processed while for example link analysis algorithms analyse high-throughput offline graphs [1].

graph databases offer an extremely flexible data model, and a mode of delivery aligned with today's agile software delivery practices [?]. - tolle überleitung hier - .A deeper description of the theory behind graph databases will be given in subsection 2, aiming to connect the data model to its fields of application as well as comparing it to RDBMS. This comparison will be picked up in subsection 3, where a use case from the SQL world will be applied to Graph Database Theory, focusing in particular on Neo4j. Lastly, our findings will be stated in subsection 4 with a general conclusion.

1.2 Graph Database Theory

1.2.1 Description of data model and functionality

1.2.2 fields of application

1.2.3 CAP Theorem

1.2.4 GraphDB vs. RDBMS

1.3 Implementation with Neo4j

1.3.1 Use Case from the SQL world

1.3.2 Installation

1.3.3 modelling

1.3.4 usage, query language

1.3.5 short conclusion, summary

1.4 Reflection

1.4.1 alternative popular graphdbs

1.4.2 conclusion

reflect on advantages disadvantages with implementation references

Bibliography

- [1] Renzo Angles and Claudio Gutierrez. An introduction to graph data management. In *Graph Data Management*, 2018.
- [2] Gabriel Kuper. The logical data model : a new approach to database logic /. 09 1985.