

Schwierigkeiten bei Implementierung und Evaluation von Datenstrukturen in Datenbanksystemen

Anton Rodenwald (19)

11. Januar 2024

Fachgebiet:	Mathematik/Informatik
Wettbewerbssparte:	Jugend Forscht
Bundesland:	Niedersachsen
Wettbewerbsjahr:	2024
Thema des Projektes:	In meinem Projekt wollte ich verschiedene in Datenbanksystemen genutzte Datenstrukturen implementieren und testen. Dabei betrachtete ich auch unterschiedliche Ansätze der Datenspeicherung und für den Datenzugriff. Bei den auftretenden Schwierigkeiten musste ich Lösungen finden bzw. etwas adaptieren.
Projektbetreuerin:	Birgit Ziegenmeyer
Institution:	Schillerschule Hannover

Kurzfassung

<Text>

Inhaltsverzeichnis

1 Motivation, Fragestellung und Ziel	1
2 Hintergrund	1
2.1 Arten von Daten und passende Datenstrukturen	1
2.2 Arten von Datenbanksystemen	2
3 Vorgehen	2
3.1 BJnet und <socket.h>	2
3.2 B-Tree	2
3.3 Binary Tree	2
3.4 Binäres Sucharray	2
3.5 Graph Datenbank	2
3.6 Pointer Zugriffe Analyse	2
4 schwierigkeiten/limitationen	2
5 Messwerte	2
6 Ergebnisse/Reflektion/Deutung der Messwerte	5
6.1 Beantwortung der Forschungsfrage	5
7 fazit	5
8 ausblick interessen geweckt	5
9 Quellenangaben	6

1 Motivation, Fragestellung und Ziel

Im Rahmen des Informatik Leistungskurses meiner Schule haben wir (Mitschüler und ich) uns mit Datenbanken und Modellierung beschäftigt. Da jedoch auf die Funktionsweise einer Datenbank nicht weiter eingegangen wurde und ich auch schon von Unterschiedlichen Datenbankansätzen, genauer Relationalen und Nicht-Relationalen, gehört hatte, wollte ich mich im Rahmen meines Projektes genauer damit beschäftigen. Dazu wollte ich eine einfache Datenbank umsetzen.

Meine Forschungsfrage ist deshalb, wie die Wahl der Datenstruktur die Geschwindigkeit einer Datenbank beeinflusst und inwiefern sich eine Datenstruktur für gewisse Daten besser oder schlechter eignet.

Mein Ursprüngliches Ziel war dabei die Programmierung einer Datenbank mit den Datenstrukturen R-Tree, B-Tree, Binary-Tree und Graphen. Davon konnte ich leider aufgrund der hohen Komplexität (Stand jetzt) vieles nicht umsetzen.

2 Hintergrund

2.1 Arten von Daten und passende Datenstrukturen

Um mir einen Überblick über zu verschaffen recherchierte ich zuerst im Internet. Dabei fand ich viele Informationen über die Arten von Daten und Datenbanksystemen, allerdings nichts konkretes über Performance Vergleiche dieser [Unk23] [JPA+12]. Verschiedene Datenarten sind dabei räumliche, zeitliche, als Objekte modellierbare und stark vernetzte Daten.

Räumliche Daten sind z. B. Kartendaten (Google Maps, OpenStreetMap). Bestimmte Landmarken oder Objekte (Bäume, Mülleimer) lassen sich beispielsweise als Punkte auf einer Fläche modellieren. Es müssen dann effizient Fragen wie z. B. "Wie viele Mülleimer gibt es in diesem Park?" oder "Was sind die 3 am besten bewerteten Restaurants in der Innenstadt?" beantwortet werden. Eine dafür ausgelegte Datenstruktur ist der R-Tree.

Zeitliche Daten können z. B. die von einem Sensor gemessene Temperaturen sein. Wichtig wäre dann z. B., dass man einfach Statistiken wie Durchschnittswerte bilden oder Anomalien erkennen könnte.

Daten, die klassischerweise in relationellen Datenbanken gespeichert werden, lassen

sich durch viele, gleich aufgebaute Objekte modellieren lassen. Beispielsweise speichert ein Unternehmen über Kunden immer die gleichen Daten (Name, E-Mail, ...). Jeder Datensatz kann dann eindeutig z. B. über eine ID identifiziert. Eine dafür geeignete Datenstruktur ist der B-Tree, eine dem Binary-Tree ähnliche Datenstruktur.

Wenn zwischen einzelnen Daten wie z. B. jedem Nutzer einer Social Media Plattform allerdings viele Beziehungen bestehen (Follower, Likes), so eignet sich ein Graph als Datenstruktur besser.

2.2 Arten von Datenbanksystemen

Moderne Datenbanksysteme ermöglichen meist, mit einem System verschiedene Speicherungsformen zu nutzen. Trotzdem muss man sich bei einem Multi-modalen System natürlich überlegen, welche Daten man nun in welcher Art von Datenstruktur speichern will. Eine Übersicht gibt es auf der Website db engines.

3 Vorgehen

3.1 BJnet und <socket.h>

3.2 B-Tree

3.3 Binary Tree

3.4 Binäres Sucharray

3.5 Graph Datenbank

3.6 Pointer Zugriffe Analyse

4 schwierigkeiten/limitationen

kein echtes server testing, weil hardware nicht so geeignet AMDuProf hat nicht funktioniert, weil keine ahnung

5 Messwerte

Meine Messungen führte ich auf meinem Desktop PC und meinem Laptop (Plugged-In) durch. Auf jedem Computer nahm ich 2 Messreihen auf. In einer Messreihe werden dabei die 4 Unterschiedlichen Anfragen an die 2 Datenstrukturen jeweils in einer Schleife direkt hintereinander durchgeführt und gemessen.

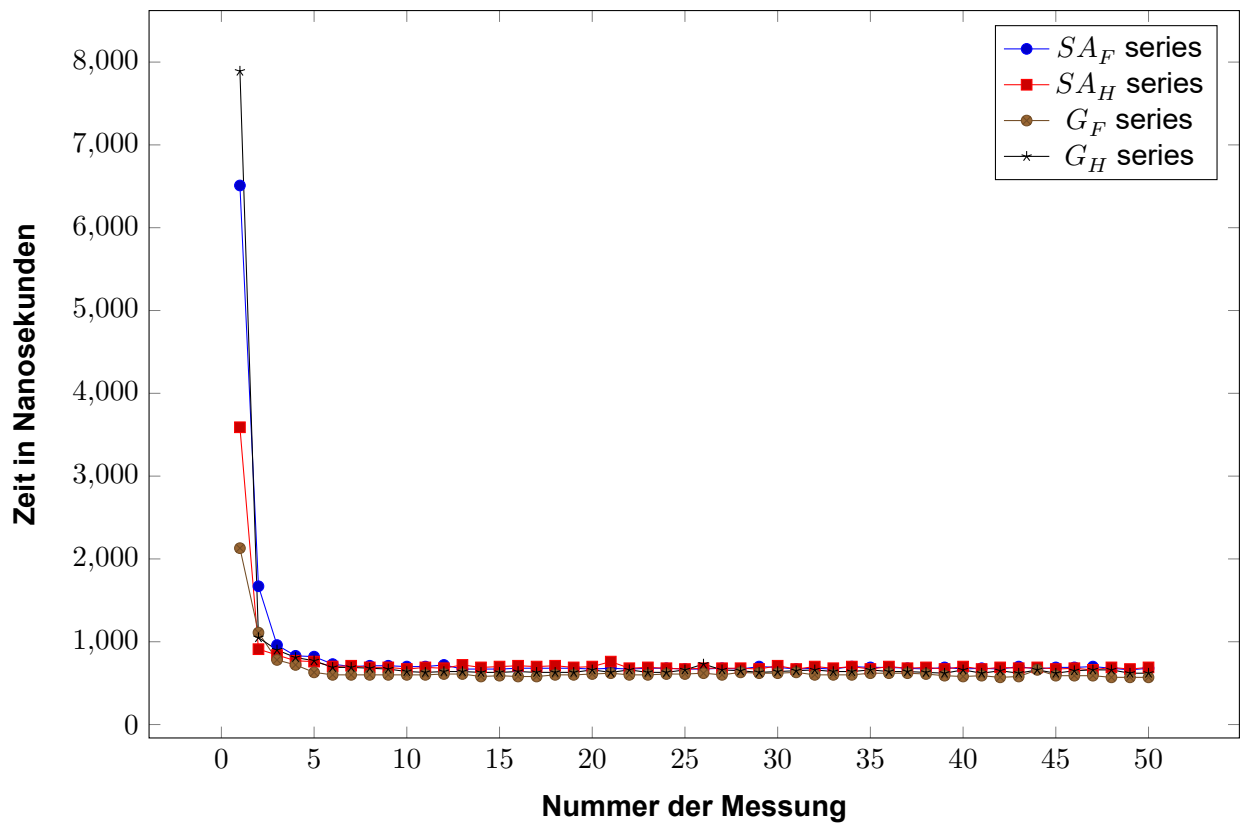


Abbildung 1: 1. Messreihe am PC (Ryzen 7 2700)

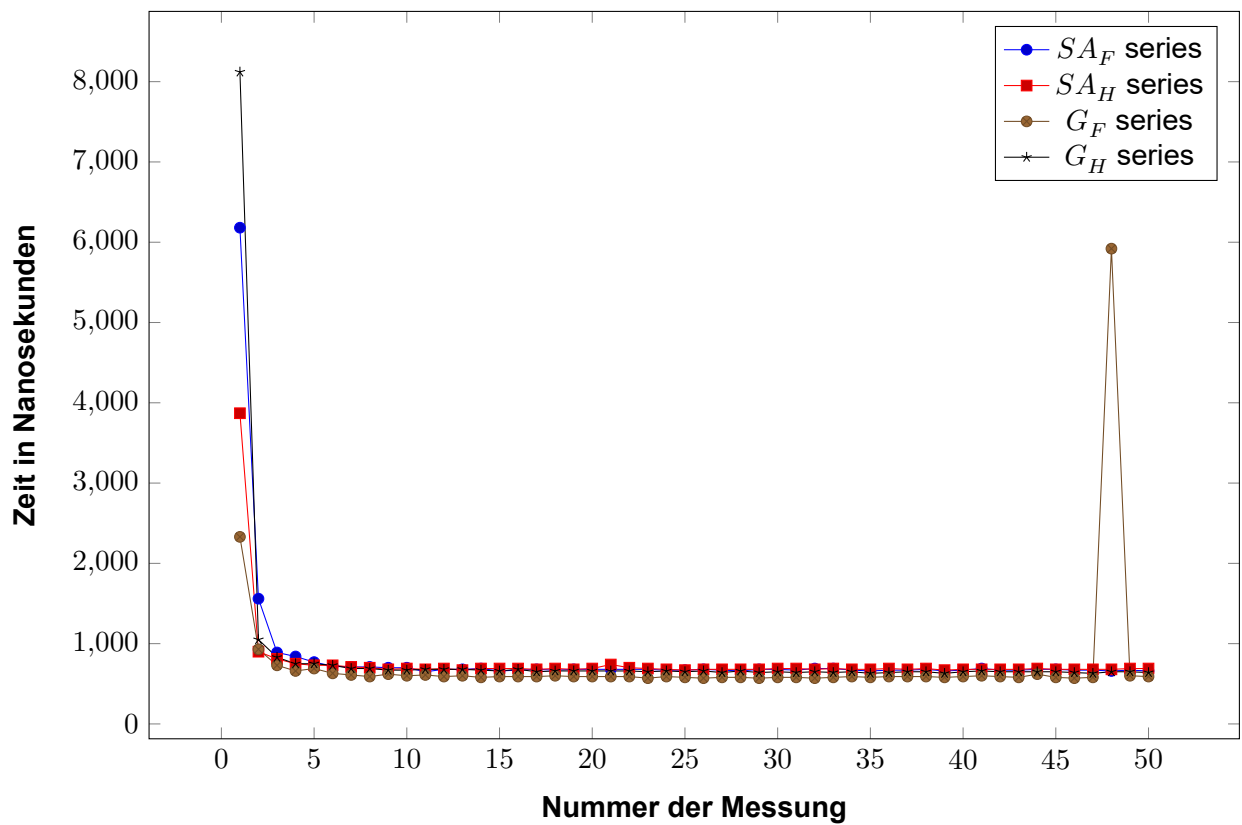


Abbildung 2: 2. Messreihe am PC (Ryzen 7 2700)

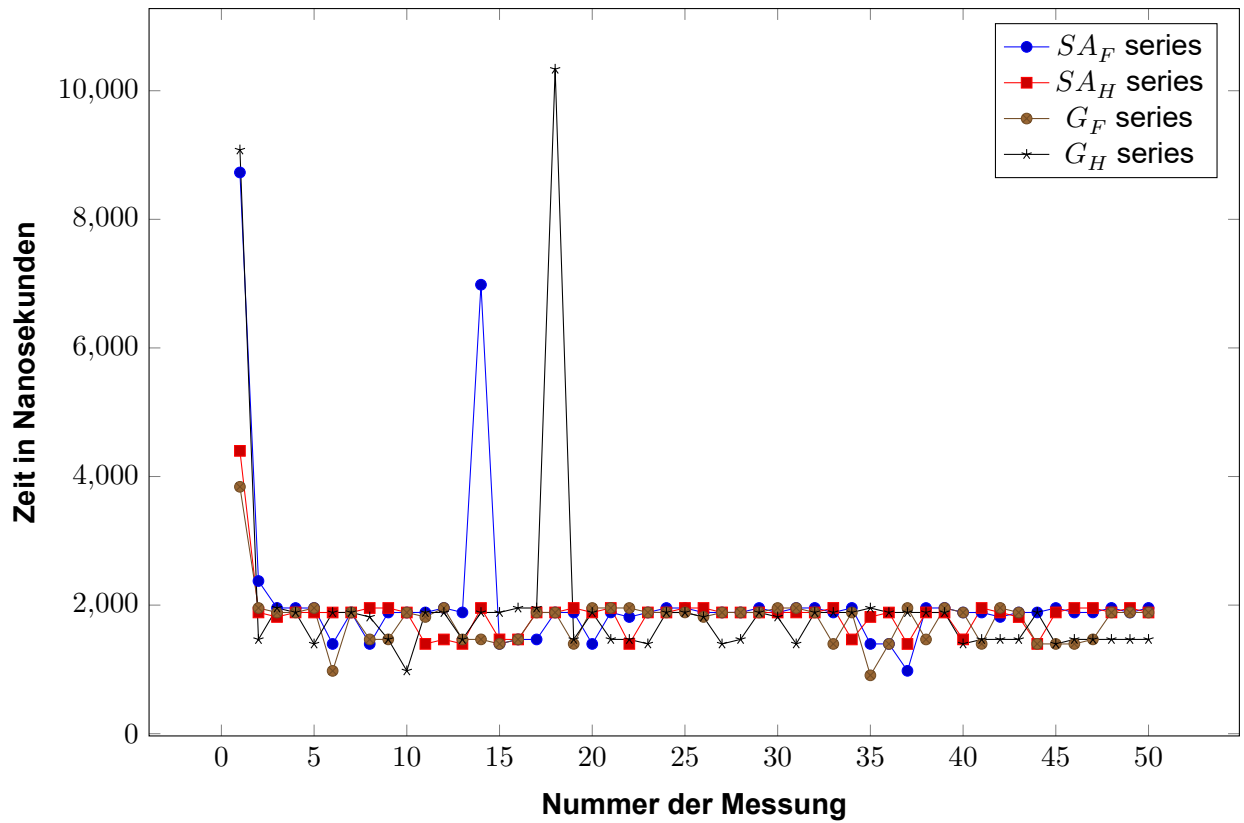


Abbildung 3: 1. Messreihe am Laptop (Ryzen 5 5500U)

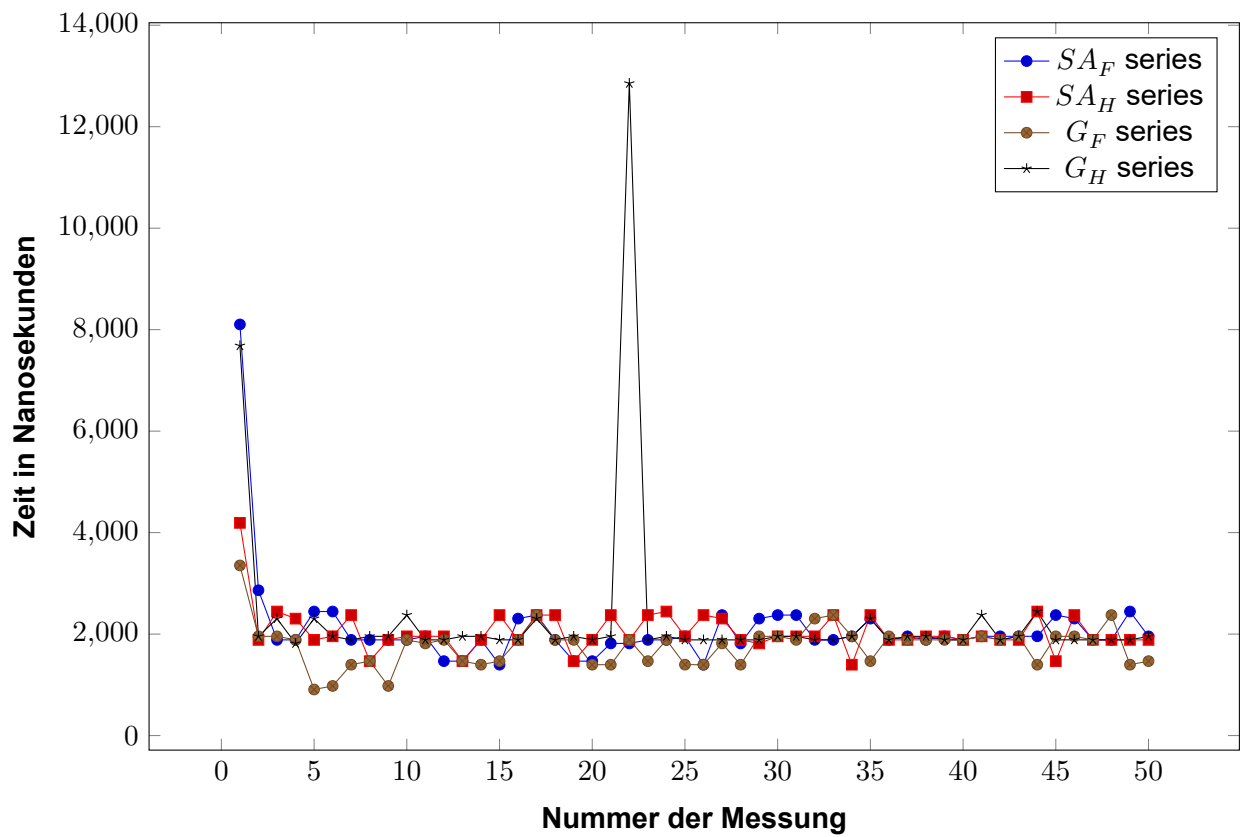


Abbildung 4: 2. Messreihe am Laptop (Ryzen 5 5500U)

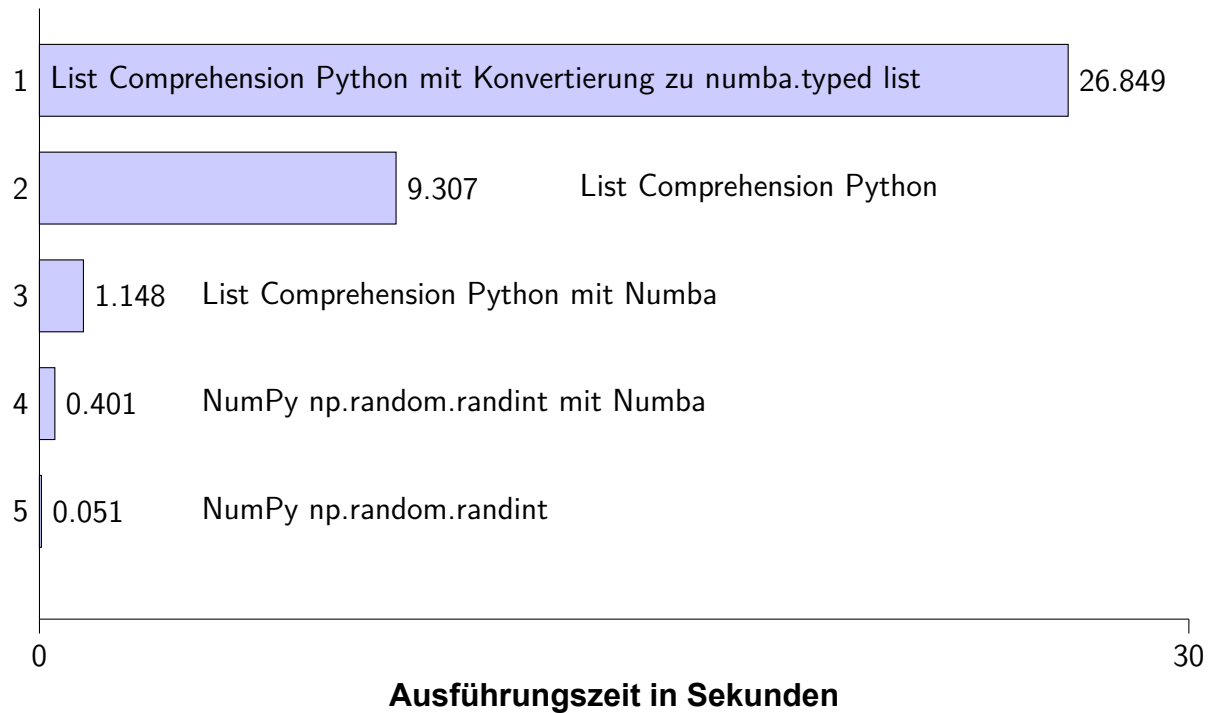
6 Ergebnisse/Reflektion/Deutung der Messwerte

6.1 Beantwortung der Forschungsfrage

7 fazit

gelernt gdb C/C++ experience mit gdb b tree socket api linux

8 ausblick interessen geweckt



[BM70] [JPA+12] [Hip08] [Unk24]

9 Quellenangaben

<https://beej.us/guide/bgnet/html/>

Literaturverzeichnis

- [BM70] R. Bayer und E. McCreight. „Organization and Maintenance of large ordered Indices“. In: (1970). URL: <https://dl.acm.org/doi/10.1145/1734663.1734671>.
- [Hip08] D. Richard Hipp. *How SQL Database Engines Work*. 2008. URL: https://www.youtube.com/watch?v=Z_cX3bzkExE (besucht am 28. 12. 2023).
- [JPA+12] Nishtha Jatana u. a. „A Survey and Comparison of Relational and Non-Relational Database“. In: (2012).
- [Unk23] Unknown. *NoSQL*. 2023. URL: <https://de.wikipedia.org/wiki/NoSQL> (besucht am 27. 12. 2023).
- [Unk24] Unknown. *AVL tree*. 2024. URL: https://en.wikipedia.org/wiki/AVL_tree.