# Horizontale Skalierungsstrategien für verteilte Datenbanksysteme in Cloud-Umgebungen: Eine vergleichende Analyse

**Clara Morrissey**
Otto-Friedrich Universität Bamberg
96049 Bamberg, Germany
clara-doreen.morrissey@stud.uni-bamberg.de

Degree: B.Sc. AI
Matriculation #: 2047695

## 1 Einleitung

Mit der zunehmenden Menge an Daten und der Notwendigkeit, diese effizient zu verwalten und zu verarbeiten, stehen Unternehmen vor der Herausforderung, ihre Datenbanksysteme zu skalieren. Horizontale Skalierung, auch bekannt als Scale-Out, bietet eine Lösung, indem sie Daten auf mehrere Server verteilt, um so die Last zu verringern und die Verfügbarkeit zu erhöhen. Im Vergleich zur vertikalen Skalierung, bei der die Kapazität eines einzelnen Servers durch Hinzufügen von mehr CPU, RAM oder Speicher erweitert wird, bietet die horizontale Skalierung mehrere Vorteile. Sie ermöglicht eine nahezu unbegrenzte Erweiterung der Kapazität, erhöht die Ausfallsicherheit durch Redundanz und ist oft kosteneffizienter, da kostengünstige Standardhardware verwendet werden kann. Vor allem für Cloud-Anwendungen ist die horizontale Skalierbarkeit von besonderer Relevanz, da Cloud-Anwendungen sich durch Hochverfügbarkeit, Elastizität und Skalierbarkeit asuzeichnen.

## 2 Konzeptuelle Grundlagen

- Datenbankensysteme "Datenbanksysteme ermöglichen die integrierte Speicherung von großen Datenbeständen, auf die mehrere Anwendungen gleichzeitig zugreifen können. Hierbei garantiert das Prinzip der Datenunabhängigkeit die weitestgehende Unabhängigkeit der Datenrepräsentation von Optimierung und Änderung der Speicherstrukturen. Sie ermöglicht auch eine Reaktion auf Änderungen der Anwendungsanforderungen, ohne die logische Struktur der Daten ändern zu müssen. Diese allgemeinen Anforderungen stellen zusammen genommen hohe Anforderungen an die interne Realisierung von Datenbanksystemen." Heuer et al. (2019)
- Skalierbarkeit "Mit Skalierbarkeit beschreiben wir die Fähigkeit eines Systems, steigende Belastungen verkraften zu können. Allerdings ist das kein eindimensionales Etikett, das wir einem System anheften können: Es ist sinnlos zu sagen »X ist skalierbar« oder »Y lässt sich nicht skalieren«. Vielmehr geht es bei einer Diskussion über Skalierbarkeit darum, Fragen zu klären wie zum Beispiel: »Wenn das System in bestimmter Art und Weise wächst, welche Optionen haben wir, um mit dem Wachstum klarzukommen?« oder »Mit welchen zusätzlichen rechentechnischen Ressourcen können wir die Mehrbelastung verarbeiten?«" Kleppmann (2019)
- Cloud-Computing Cloud Computing ermöglicht den bedarfsorientierten Zugriff auf IT-Ressourcen über das Internet. Diese Ressourcen umfassen Anwendungen, physische und virtuelle Server, Speicherplatz, Entwicklungstools, Netzwerkfunktionen und mehr. Die Ressourcen werden in einem entfernten Rechenzentrum gehostet, das von einem Cloud-Dienstanbieter (Cloud Service Provider, CSP) verwaltet wird. Der CSP stellt diese Dienste

entweder gegen eine monatliche Abonnementgebühr oder auf Basis des tatsächlichen Verbrauchs zur Verfügung.

## 3 Zielsetzung und Forschungsfrage

- Zielsetzung
  Die Wahl der richtigen Skalierungsstrategie hängt von vielen Faktoren ab, darunter die Art der Datenbank (relational oder NoSQL), die spezifischen Anforderungen der Anwendung und die Infrastruktur des Unternehmens. Es gibt eine Vielzahl von Implementierungen und Techniken für horizontale Skalierung, wie Sharding, Replikation und Partitionierung, die je nach Datenbanktyp und -anbieter unterschiedlich umgesetzt werden. Diese Arbeit zielt darauf ab, die verschiedenen Methoden der horizontalen Skalierung für relationale und NoSQL-Datenbanksysteme zu untersuchen, deren Vor- und Nachteile darzustellen und die Performance von Sharding-Implementierungen bei ausgewählten Datenbankanbietern (PostgreSQL mit Citus, CockroachDB, MongoDB und Amazon DynamoDB) zu testen und zu vergleichen. Durch diese vergleichende Analyse sollen Empfehlungen für den praktischen Einsatz der unterschiedlichen Skalierungsstrategien gegeben werden.
- Forschungsfragen
  1. Wie unterscheiden sich die Implementierungen der horizontalen Skalierung zwischen relationalen und NoSQL-Datenbanksystemen? Welche Techniken (z.B. Sharding, Replikation, Partitionierung) werden von relationalen Datenbanken wie PostgreSQL (mit Citus) und CockroachDB verwendet? Welche Techniken werden von NoSQL-Datenbanken wie MongoDB und Amazon DynamoDB verwendet?
  2. Welche Vor- und Nachteile bieten die verschiedenen Methoden der horizontalen Skalierung in den untersuchten Datenbanksystemen? Wie wirken sich die verschiedenen Skalierungstechniken auf die Datenkonsistenz, Verfügbarkeit und Partitionstoleranz aus? Welche Herausforderungen und Komplexitäten treten bei der Implementierung und Verwaltung der Skalierungstechniken auf?
  3. Wie verhalten sich die Performance-Charakteristika der ausgewählten Datenbanksysteme unter unterschiedlichen Workloads? Wie unterscheiden sich Latenz und Durchsatz zwischen den Datenbanksystemen bei leseintensiven, schreibintensiven und gemischten Workloads? Wie gut skalieren die Systeme bei zunehmender Daten- und Lastmenge?
  4. Welche praktischen Empfehlungen können aus den Testergebnissen für den Einsatz von horizontaler Skalierung in relationalen und NoSQL-Datenbanksystemen abgeleitet werden? Welche Datenbanksysteme und Skalierungstechniken eignen sich am besten für spezifische Anwendungsfälle und Anforderungen? Welche Best Practices können zur Optimierung der Performance und Effizienz der horizontalen Skalierung angewendet werden?

## 4 Methodik

- Literaturrecherche: Untersuchung aktueller Forschungsarbeiten und Dokumentationen zu horizontaler Skalierung und verteilten Datenbanksystemen.
- Experimentelle Evaluation: Durchführung von Performance-Tests in einer kontrollierten Umgebung mit ausgewählten Datenbankanbietern.
- Datenanalyse: Auswertung und Vergleich der gesammelten Daten zur Bewertung der Effizienz und Skalierbarkeit der verschiedenen Implementierungen.

## 5 Gliederung

1. Einführung in die horizontale Skalierung
   - Definition und Grundlagen
   - Vergleich zu vertikaler Skalierung
   - Anwendungsbereiche und Relevanz in der modernen IT-Landschaft
2. Konzepte der horizontalen Skalierung

- Relationale Datenbanken
  - Sharding
  - Replikation
  - Partitionierung
- NoSQL-Datenbanken
  - Sharding
  - Replikation
  - Consistent Hashing

3. Vor- und Nachteile der horizontalen Skalierung
   - Relationale Datenbanken
     - Vorteile: ACID-Transaktionen, Datenintegrität, SQL-Unterstützung
     - Nachteile: Komplexität bei der Skalierung, Konsistenzprobleme
   - NoSQL-Datenbanken
     - Vorteile: Hohe Skalierbarkeit, Flexibilität, einfache Datenmodelle
     - Nachteile: Eventual Consistency, eingeschränkte Transaktionen, fehlende SQL-Unterstützung Implementierungen bei verschiedenen Datenbankanbietern
4. Implementierung bei verschiedenen Datenbankanbietern
   - Relationale Datenbanken
     - PostgreSQL (mit Citus): Horizontale Partitionierun via Citus
     - CockroachDB: Globale Verteilung und Konsistenz
   - NoSQL-Datenbanken
     - MongoDB: Dokumentenorientierte Skalierung und Sharding
     - Amazon DynamoDB: Schlüssel-Wert- und dokumentenorientierte Datenbank mit automatischer Skalierung
5. Performance-Tests
   - Testumgebung und -setup
     - Metriken: Latenz, Durchsatz, Konsistenz
     - Vergleich der Performance bei unterschiedlichen Workloads (Lese-intensiv, Schreib-intensiv, gemischt)
     - Analyse der Ergebnisse
6. Diskussion
   - Interpretation der Testergebnisse
   - Praktische Empfehlungen für den Einsatz verschiedener Skalierungsmethoden
   - Grenzen der horizontalen Skalierung
   - Fazit
7. Zusammenfassung der Erkenntnisse
   - Ausblick auf zukünftige Entwicklungen im Bereich der verteilten Datenbanksysteme

## 6 weiterführende Literaturrecherche

- Cost-aware horizontal scaling of NoSQL databases using probabilistic model checking. Naskos et al. (2017)
- An automated implementation of hybrid cloud for performance evaluation of distributed databases Mansouri et al. (2020)
- Benchmarking scalability and elasticity of distributed database systems Mansouri et al. (2020)
- Die neue Realität: Erweiterung des Data Warehouse um Hadoop, NoSQL and Co. Müller (2014)
- Vergleich und Evaluation zwischen modernen und traditionellen Datenbankkonzepten unter den Gesichtspunkten Skalierung, Abfragemöglichkeit und Konsistenz Petersohn (201)

## References


A. Heuer, G. Saake, and K.-U. Sattler. *Datenbanken : Implementierungstechniken*. MITP Verlags GmbH I& Co. KG, 2019.

M. Kleppmann. *Datenintensive Anwendungen designen*. Dpunkt.verlag, 2019.

Y. Mansouri, V. Prokhorenko, and M. A. Babar. An automated implementation of hybrid cloud for performance evaluation of distributed databases. *Journal of Network and Computer Applications*, 167:102740, 2020. ISSN 1084-8045. doi: https://doi.org/10.1016/j.jnca.2020.102740. URL `https://www.sciencedirect.com/science/article/pii/S1084804520302149`.

S. Müller. Die neue realität: Erweiterung des data warehouse um hadoop, nosql i& co. *HMD Praxis der Wirtschaftsinformatik*, 51(4):447–457, aug 2014. ISSN 2150-8097. doi: 10.14778/2732977.2732995. URL `https://doi.org/10.1365/s40702-014-0053-9`.

A. Naskos, A. Gounaris, and P. Katsaros. Cost-aware horizontal scaling of nosql databases using probabilistic model checking. *Cluster Comput*, 20:2687–2701, 2017.

N. Petersohn. *Vergleich und Evaluation zwischen modernen und traditionellen Datenbankkonzepten unter den Gesichtspunkten Skalierung, Abfragemöglichkeit und Konsistenz*. Diplomica Verlag, 201.