

Nils Petersohn

**Vergleich und Evaluation zwischen
modernen und traditionellen
Datenbankkonzepten unter den
Gesichtspunkten Skalierung,
Abfragemöglichkeit und Konsistenz**

Bibliografische Information der Deutschen Nationalbibliothek:

Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de/> abrufbar.

Dieses Werk sowie alle darin enthaltenen einzelnen Beiträge und Abbildungen sind urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsschutz zugelassen ist, bedarf der vorherigen Zustimmung des Verlanges. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen, Auswertungen durch Datenbanken und für die Einspeicherung und Verarbeitung in elektronische Systeme. Alle Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

Copyright © 2010 Diplomica Verlag GmbH
ISBN: 9783842806085

Nils Petersohn

Vergleich und Evaluation zwischen modernen und traditionellen Datenbankkonzepten unter den Gesichtspunkten Skalierung, Abfragemöglichkeit und Konsistenz

Nils Petersohn

Vergleich und Evaluation zwischen modernen und traditionellen Datenbankkonzepten unter den Gesichtspunkten Skalierung, Abfragemöglichkeit und Konsistenz

Nils Petersohn

Vergleich und Evaluation zwischen modernen und traditionellen Datenbankkonzepten unter den Gesichtspunkten Skalierung, Abfragemöglichkeit und Konsistenz

ISBN: 978-3-8428-0608-5

Herstellung: Diplomica® Verlag GmbH, Hamburg, 2010

Zugl. Fachhochschule Brandenburg, Brandenburg, Deutschland, Bachelorarbeit, 2010

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtes.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Die Informationen in diesem Werk wurden mit Sorgfalt erarbeitet. Dennoch können Fehler nicht vollständig ausgeschlossen werden und der Verlag, die Autoren oder Übersetzer übernehmen keine juristische Verantwortung oder irgendeine Haftung für evtl. verbliebene fehlerhafte Angaben und deren Folgen.

© Diplomica Verlag GmbH

<http://www.diplomica.de>, Hamburg 2010

Inhaltsverzeichnis

1. Einleitung	1
1.1. Motivation	1
1.2. Zielsetzung	2
1.3. Überblick	2
2. Konzepte moderner Datenbanken	4
2.1. Consistent Hashing	5
2.2. Brewer's Theorem und "letztendliche Konsistenz"	8
2.3. Versionierung der Daten	13
2.4. MapReduce	14
2.4.1. Map	16
2.4.2. Reduce	16
2.4.3. Abarbeitungsübersicht	17
2.5. Zusammenfassung	19
3. Riak - ein Key-Value-Store	20
3.1. Key-Value-Store	20
3.2. Riak	20
3.3. Das konsistente Hashverfahren und die CAP-Parameter	21
3.4. eventually consistent	26
3.5. MapReduce	28
3.6. Zusammenfassung	30
4. Relationale Datenbanken	31
4.1. Skalierung und Partitionierung	31
4.1.1. vertikal	32
4.1.2. horizontal	32
4.1.3. funktionale Partitionierung	32
4.1.4. Sharding	33
4.1.5. Abfragen über mehrere Shards	33
4.2. Zusammenfassung	34
5. Gegenüberstellungen	35
5.1. Abfragedynamik vs. Daten-Zuwachsrate	35
5.1.1. geringe Abfragedynamik	35
5.1.2. mittlere Abfragedynamik	36
5.1.3. mittlere bis hohe Datendynamik und vorhersehbare Daten-Zuwachsrate	36
5.1.4. hohe Abfragedynamik und hohe Daten-Zuwachsrate	37

5.1.5. Zusammenfassung und Bewertung	37
5.2. Skalierung und Konsistenz	37
5.3. Abfragemöglichkeiten	41
5.3.1. Anwendungsfälle für MapReduce	43
5.3.2. Anwendungsfälle für SQL	47
5.4. Abfragen: Riak und MySql	48
5.4.1. Stored Procedures	48
5.4.2. Von MySQL zu Riak	50
5.4.3. MapReduce	52
5.4.4. Bewertung	55
6. Zusammenfassung	56
Literaturverzeichnis	58
A. Anhang	63
Anhang	63
A.1. MySQL Stored Procedure - Verfügbarkeitsabfragen im Hotel System mit Stored Procedures	63
A.2. Riak MapReduce - Verfügbarkeitsabfragen im Hotel System mit MapReduce	64
A.3. Amazon EC2 Riak Installation Script und Testfall	66
A.4. Amazon EC2 Riak Join Script	71

1. Einleitung

Zehntausende Web-Services verwenden relationale Datenbanken, um Daten zu speichern und auszulesen. Im Vergleich zu modernen Konzepten können relationale Datenbanken als wichtigster Stellvertreter für “traditionelle Technologien” bezeichnet werden. Wenn man als Entwickler zu Seiten wie Google.com, Facebook.com, Amazon.com, Digg.com, Ebay.com, Yahoo.com, Twitter.com, oder Dawanda.com surft, wird meist angenommen, dass eine verteilte relationale Datenbank verwendet wird. Die Annahme ist zu 50% richtig, jedoch ist die Datenhaltung meist nicht relational. Diese Großunternehmen verwalten mehrere hundert Gigabytes, bis hin zu 100.000 Gigabyte an Daten, und mussten in den letzten sechs Jahren Lösungen finden, um erfolgreich diese riesigen Datenmengen zu beherrschen. Google erfand vor ca. sieben Jahren ein Verfahren, um Datenmengen im Petabytebereich zu beherrschen. Facebook entwickelte selbst eine Datenbanktechnologie, um die Posteingänge von Benutzern verfügbar zu machen, Twitter.com adaptiert diese Technologie für andere Zwecke (vgl. [Lai10]). Amazon.com entwickelte “Dynamo”, um Hochverfügbarkeit für deren weltgrößte E-Commerce Plattform zu schaffen. Diese und andere Eigenentwicklungen entstanden aus der Notwendigkeit heraus, riesige Datenmengen bzw. Datenbanken hoch verfügbar, konsistent und skalierbar zu machen.

1.1. Motivation

Seit den letzten drei Jahren sind alternative “Open-Source-Implementierungen” dieser Entwicklungen entstanden. Die Veröffentlichung der Konzepte und Technologien führten zu einer ganzen Bewegung namens “NoSQL”. Sind diese Konzepte vorteilhafter, um eine bessere und für Entwickler einfachere Skalierung, Abfragemöglichkeit und Datenkonsistenz in einem hochverfügbaren Datenbanksystem, zu gewährleisten? Wie werden komplexe Abfragen in modernen und traditionellen verteilten Systemen gemacht und wie werden diese ausgeführt? Speziell stellt sich die Frage, ob das MapReduce Verfahren ein vollständiger Ersatz für SQL ist. Für welche Einsatzzwecke sind beide besonders gut geeignet und für welche weniger?