

FOM MÜNCHEN

FACHBEREICH WIRTSCHAFTSINFORMATIK

Seminararbeit

Optimierung von MySQL Anfragen unter Zuhilfenahme von Explain

Eingereicht von:

Oliver Kurmis

Perfallstraße 8 81675 München Email: oliver@kurmis.com

Abgegeben am:

1. Juli 2014

Erarbeitet im:

3. Semester

Inhaltsverzeichnis

Abkürzungsverzeichnis	III
Abbildungsverzeichnis	III
Tabellenverzeichnis	III
1. Einleitung	1
1.1. Problemstellung	1
1.2. Zielsetzung der vorliegenden wissenschaftlichen Auseinandersetzung . .	1
1.3. Vorgehensbeschreibung	1
2. Theoretische Grundlagen	1
2.1. Der physische Zugriff auf die Daten	1
2.2. Speicherstrukturen	2
2.3. Bearbeitung von SQL-Statements	2
2.4. Optimierungen : frühzeitige Restriktionen, JOINS	3
3. Beispiel MySQL	3
3.1. einfache Select-Anfragen (eine Tabelle)	3
3.2. Umschreiben von Nicht-Select-Anfragen	3
3.3. Die Spalten der EXPLAIN-Ausgabe	3
3.4. EXPLAIN EXTENDED	3
3.5. EXPLAIN PARTITIONS	3
3.6. Abfragen mit mehreren Tabellen	3
3.7. Optimierungsmöglichkeiten und Benchmarking	3
3.8. Visuelles EXPLAIN (graphische Werkzeuge)	3
4. Fazit und Ausblick	3
A. Anhang	4
Literatur	5

Abkürzungsverzeichnis

CPU Central Processing Unit (deutsch: Hauptprozessor)

DB Datenbank

RAM Random Access Memory (deutsch: Hauptspeicher oder Arbeitsspeicher)

RDBMS Relationales Datenbank-Managementsystem

SQL Structured Query Language

Abbildungsverzeichnis

Tabellenverzeichnis

1. Einleitung

1.1. Problemstellung

Datenbank-Systeme finden heute in nahezu allen IT-Systemen Verwendung. Der Optimierung von Datenbank-Anfragen kommt daher eine große Bedeutung zu. Hierfür gibt es eine Vielzahl von Möglichkeiten, z.B. Latenz und Bandbreite der Anbindung der Datenbank, Leistungsfähigkeit des Datenbank-Servers, Anzahl der Datenbank-Anfragen im Programmcode, Cachingmechanismen.

Hat man andere Flaschenhälse ausgeschlossen oder bereits optimiert, gilt es die relevanten SQL-Abfragen des Systems zu identifizieren und gezielt zu optimieren.

Viele RDBMS stellen mit dem SQL-Kommando EXPLAIN eine Möglichkeit zur Verfügung, mehr über die innere Arbeitsweise der Datenbank bei einer bestimmten SQL-Abfrage zu erfahren. Durch gezielte Veränderung der SQL-Abfrage oder des Datenschemas kann somit die Bearbeitung der Abfrage optimiert werden.

1.2. Zielsetzung der vorliegenden wissenschaftlichen Auseinandersetzung

Die folgende Arbeit bezieht sich speziell auf die Optimierung von SQL-Anfragen mittels EXPLAIN bei dem RDBMS MySQL. Es soll untersucht werden

1.3. Vorgehensbeschreibung

2. Theoretische Grundlagen

2.1. Der physische Zugriff auf die Daten

HDD-Blockgerätetreiber-Dateisystemtreiber-DBMS

2.2. Speicherstrukturen

Binärbaum, B-Baum, Hashing, Heap

2.3. Bearbeitung von SQL-Statements

Umsetzung in relationale Algebra

2.4. Optimierungen : frühzeitige Restriktionen, JOINS

3. Beispiel MySQL

3.1. einfache Select-Anfragen (eine Tabelle)

3.2. Umschreiben von Nicht-Select-Anfragen

3.3. Die Spalten der EXPLAIN-Ausgabe

3.4. EXPLAIN EXTENDED

3.5. EXPLAIN PARTITIONS

3.6. Abfragen mit mehreren Tabellen

3.7. Optimierungsmöglichkeiten und Benchmarking

3.8. Visuelles EXPLAIN (graphische Werkzeuge)

4. Fazit und Ausblick

Beschränkungen! Optimierung wichtig Mit Explain möglich nicht immer exakte Angaben Kontrolle der Optimierung mit Benchmarks nötig möglichst bereits in den Entwicklungsprozess integrieren, und nicht erst wenn es brennt

A. Anhang

Was so alles in einen Anhang kommt

Literatur

- [1] Schwartz, B., Zaitsev, P., Tkachenko, V., Zawodny, J.D.,Lentz, A., Balling, D.J. (2009) *High Performance MySQL. Optimierung, Datensicherung, Replikation & Lastverteilung*, 2. Auflage, O'Reilly Verlag, 2009

- [2] Sauer, H. (1998) *Relationale Datenbanken, Theorie und Praxis*, 4. Auflage, Addison Wesley Longman Verlag, 1998

- [3] Bradford, R. (2011) *Effective MySQL: Optimizing SQL Statements*, McGraw-Hill Osborne Media, 2011