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

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, Cachingmeachanismen.

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

-Literaturrecherche betrieben, -wesentliche Punkte zusammengefasst -an Beispieldatenbank experimentell nachvollzogen

2 Theoretische Grundlagen

2.1 Der physische Zugriff auf die Daten

Daten einer DB werden in der Regel auf einer Festplatte (HDD) oder einem Flash-Laufwerk (SSD) gespeichert. Das RDBMS nutzt dazu Funktionen des Betriebssystems auf verschiedenen Ebenen. Dateisystem-Treiber, nimmt Lese und Schreibanforderungen für Datensätze an und gibt rechnet diese in nie durchnumerierten Blöcke des Blockgerätes um. Der Blockgeräte-Treiber liest dann die entsprenden Blöcke von der Platte oder schreibt sie.

DBMS -> Dateisystemtreiber -> Blockgerätetreiber -> HDD/SSD

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

As of MySQL 5.6.3, permitted explainable statements for EXPLAIN are SELECT, DELETE, INSERT, REPLACE, and UPDATE. Before MySQL 5.6.3, SELECT is the only explainable statement. [4]

3.3 Die Spalten der EXPLAIN-Ausgabe

id identifiziert das SELECT, zu dem die Zeile gehört -bei einfachen SELECT immer nur 1

select_type einfaches oder komplexes SELECT -SIMPLE = Einfaches select, keine Unterabfragen oder UNIONS -PRIMARY = äußeres eines komplexen SELECT -SUBQUERY = Select in einer Unterabfrage -DERIVED = SELECT in Unterabfrage in FROM-Klausel -UNION = zweites bzw. nachfolgende SELECT einer Union -UNION RESULT = Ergebnis des UNION, wird aus temporärer Tabelle geholt.

table auf welche Tabelle wird zugegriffen -Tabellenname oder Alias -Spalte von oben nach unten lesen

type: Wie ist der Zugriffstyp, wie wird MySQL die Zeilen in der Tabelle auffinden? -ALL = Tablescan, Tabelle muss in der Regel von Anfang bis Ende durchlaufen werden -index = Wie Tablescan, aber in Indexreihenfolge. Sortierung wird vermieden. -range=Bereichsscan= eingeschränkter Indexscan, z.B bei BETWEEN oder WHERE x > -ref = Indexzugriff, Index-Lookup, Zeilen entsprechen einem Wert, nur bei einem nichteindeutigen Index, Index wird mit einem Referenzwert verglichen. Variante: ref_or_null -eq_ref = Index-Lookup mit eindeutigem Treffer, bei Primärschlüssel oder

eindeutigem Index -const,system = konnte von MySQL wegoptimiert oder in eine Konstante umgewandelt werden. -NULL = Abfrage kann von MySQL bei der Optimierung aufgelöst werden, kein Zugriff auf Tabelle oder Index, z.B. Minimum einer indizierten Spalte

possible_keys: -welche Indizes kömnnten benutzt werden -frühe Phase der Optimierung -können später nutzlos sein

key: -welchen Index wählt der Optimierer für die Abfrage -auch abdeckende Indizes

key_len -wie viel Byte (Spaltenbreite) eines Index werden benutzt -welche Spalten des Index werden genutzt, von links beginnend

ref -welche Spalten aus früheren Tabellen werden benutzt, um in dem key-Index nachzuschlagen

rows -Schätzung, Anzahl der Zeilen, die gelesen werden müssen -pro schleife im Nested-Loop-Join-Plan -Schätzung kann ungenau sein

filtered -neu seit MySQL 5.1 -bei Explain EXTENDED -pessimistische Schätzung des Prozentsatzes der Zeilen

Extra -Using Index = abdeckender Index benutzt -Using where = Zeilen nachträglich gefiltert, also nicht den Index für die WHERE-Bedingung genutzt -Using temporary = temporäre Tabelle für Sortierung -Using filesort = externe Sortierung, im RAM oder auf HDD -ranke checked for each record (index map: N) = kein guter Index vorhanden EXPLAIN Output Format [5]

3.3.1 EXPLAIN EXTENDED

3.3.2 EXPLAIN PARTITIONS

3.4 Abfragen mit mehreren Tabellen

3.5 Optimierungsmöglichkeiten und Benchmarking

3.6 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

5 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

[4] *Optimizing Queries with EXPLAIN* http://dev.mysql.com/doc/refman/5.6/en/using-explain.html

[5] *EXPLAIN Output Format* http://dev.mysql.com/doc/refman/5.6/en/explain-output.html