

# MySQL kann auch NoSQL

Erkan Yanar

Frankfurt am Main

**Schlüsselworte:** MySQL, NoSQL

## Einleitung

MySQL ist die (relationale) Datenbank des Internets mit der meisten Verbreitung. Seit kürzerer Zeit gibt es einen Hype mit dem Namen NoSQL. Dieser meint – in verschiedensten Inkarnationen - Limitierungen von relationalen Datenbanken durch neue Ansätze und Lockerungen von Constraints beheben zu wollen und können. In diesem Vortrag wird es nicht auf den Sinn und Unsinn des NoSQL-Hypes eingegangen. Der Vortrag beschreibt Entwicklungen in und um MySQL welche als Antwort auf NoSQL gesehen werden sollten und das Anwendungsspektrum von MySQL erweitern. Damit wiederum kann sich eine Migration oder die Einführung einer neuen Technik schnell erübrigen haben.

Der Vortrag bietet einen groben Überblick über die Techniken, welche lediglich vorgestellt werden können. Zudem werden nur Lösungen vorgestellt welche Open Source sind.

Grob unterteilt ergeben sich folgende Punkte:

## KeyValue

KeyValue Stores wie z.B Memcached werden schon jetzt oft mit MySQL-Installationen verwendet. Sie bieten – aufgrund der simplen Datenspeicherung und einem kompakten Protokoll - einen einfachen und schnellen Zugang zu Daten. Im Zusammenhang mit MySQL werden hier oft Ergebnisse/Aufrufe zwischengespeichert. Dadurch bedingt, braucht die MySQL Instanz nicht mehrmals befragt zu werden.

KeyValue Implementierungen für MySQL umgehen unter anderem den SQL-Parser/Optimizer um Anfragen direkt an die StorageEngine weiter zu leiten. Denn - wie in Abb 1 zu sehen - verbraucht eine Query die meiste Zeit außerhalb der StorageEngine. Indem die Getter und Setter eben diesen Overhead umgehen, ist eine vielfach höhere Anzahl der Queries/s möglich. Zum einen konkurrieren dies Lösungen in Puncto Performance mit Lösungen wie Memcached, sicherlich spart man sich mit diesen Lösungen – welche eine separate Installation einer NoSQL-KeyValue Instanz überflüssig machen – Roundtrips, Dateninkonsistenz verschiedener Instanzen und den 'doppelten' Speicherverbrauch.

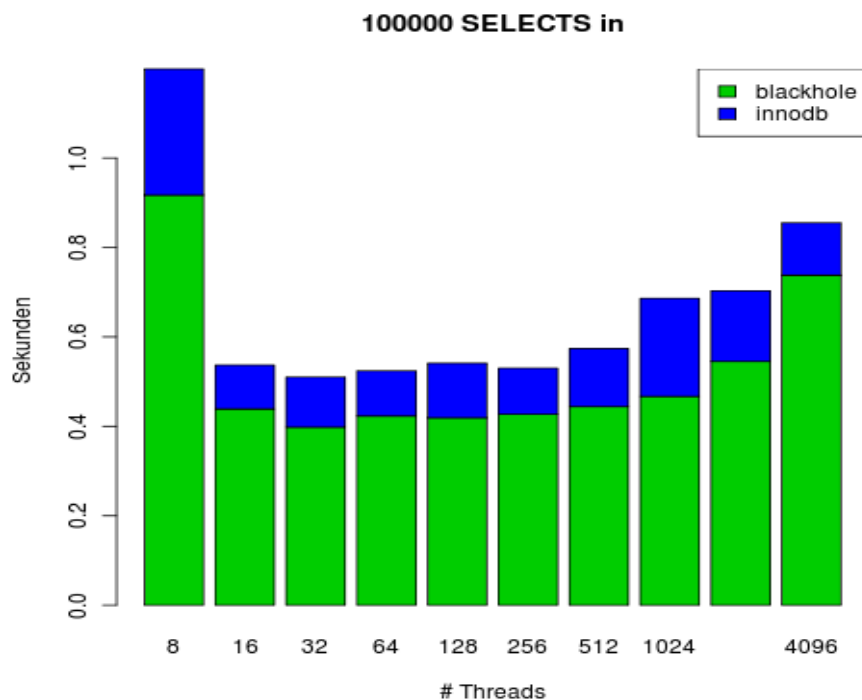


Abb 1: Die Zeit welche für 10000 SELECTs verbraucht wird. Der grüne Balken ist dem Overhead außerhalb der StorageEngine gleich zu setzen.

Um MySQL als KeyValue Store nutzbar zu machen gibt es unterschiedliche Ansätze. Der MySQL-Cluster kann von grund auf via einer API angesprochen werden.

Oracle wird für den MySQL-Cluster (7.2) und den MySQL Server (5.6.x) ein Memcached-Plugin liefern. Dieses wird via dem Memcached-Protokoll angesprochen werden können.

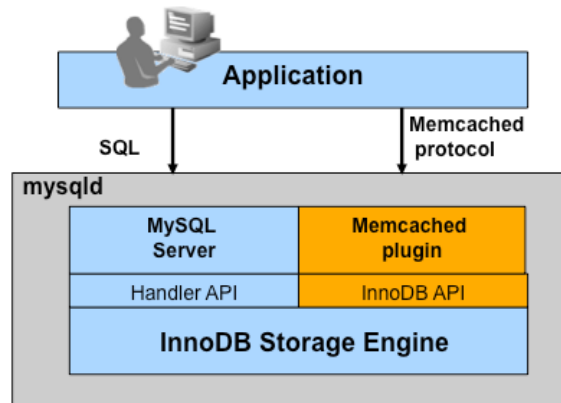


Abb2: Mit dem Memcached-Plugin nehmen die Abfragen einen eigenen Weg zur Storage Engine. Bild: <http://dev.mysql.com/tech-resources/articles/nosql-to-mysql-with-memcached.html>

MariaDB und Percona liefern das HandlerSocket-Plugin von [Yoshinori Matsunobu](#) mit. Dieses umgeht auch den Overhead des SQL-Parsers/Optimizers von MySQL und ermöglicht einen direkten Zugang zur StorageEngine. Im Gegensatz zum Memcached-Plugin implementiert HandlerSocket ein eigenes Protokoll. Dieses erlaubt einen vielfältigeren Zugang zu den Daten, als dies mit dem Memcached-Plugin möglich ist.

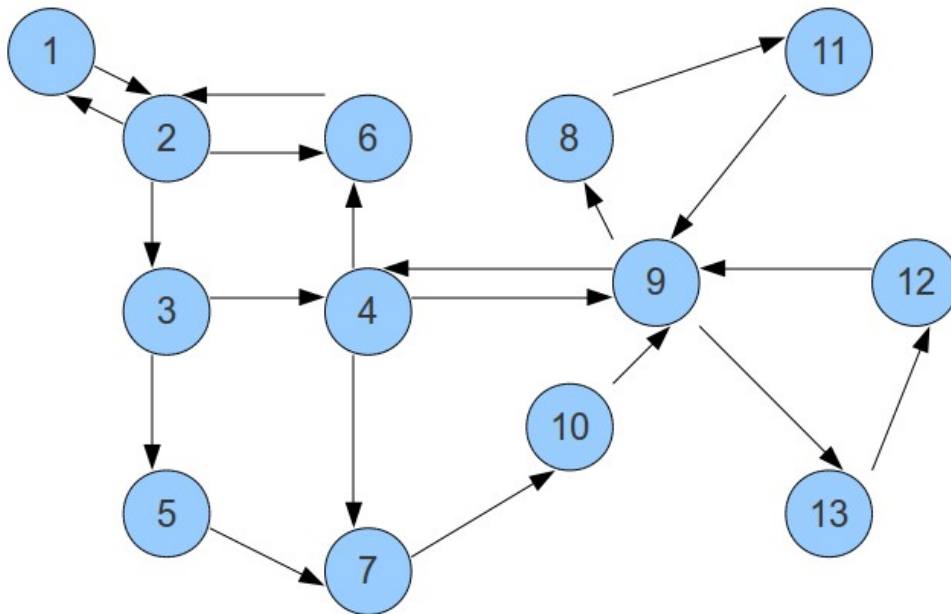
Eine weitere Möglichkeit (nur) den Optimizer zu umgehen ist die Verwendung der HANDLER Commands, welche MySQL schon lange zur Verfügung stellt.

## DocumentStore

Die entscheidende Eigenschaft von DocumentStore ist, dass jedes Dokument(Row) eine variable Anzahl von Spalten haben kann. MariaDB 5.3 führte dazu die Dynamic Columns ein. Diese erlauben es Spalten zu definieren, welche pro Row unterschiedlich viele Key:Value Paare aufnehmen können. Als Container für die Dynamic Columns dient ein Blob. Der Zugriff erfolgt über Funktionen:

```
> CREATE TABLE dyni ( id INT, sonderlocke BLOB);
> INSERT INTO dyni VALUES(1,COLUMN_CREATE(1 , "`hallo`", 2 , 42));
> SELECT id, COLUMN_LIST(sonderlocke) from dyni;
+-----+-----+
| id   | COLUMN_LIST(sonderlocke) |
+-----+-----+
|    1 | 1,2                       |
+-----+-----+
> SELECT id,COLUMN_GET(sonderlocke,1 as char) from dyni;
+-----+-----+
| id   | COLUMN_GET(sonderlocke,id as char) |
+-----+-----+
|    1 | hallo                               |
+-----+-----+
> UPDATE dyni SET sonderlocke=COLUMN_ADD(sonderlocke,2,NULL,3,"\/");
```

## Graphen



Abd2: Graphendatenbanken dienen dazu performant Verbindungen(Kanten) von Nodes(Ecken) zu berechnen.

Graphendatenbanken wie z.B. Neo4J und Sones setzen da an, wo klassische Konzepte von SELF-Joins nicht mehr skalieren können. Mit OQGraph existiert für MySQL eine StorageEngine welche auch die Daten graphenfreundlich speichert. Die OQGraph StorageEngine gehört zum Lieferumfang von MariaDB.

## Messages

RabbitMQ ist eine NoSQL Lösung für Messages. Dem sicheren übertragen von 'Nachrichten'. Mit Q4M (von Kazuho Oku) existiert eine StorageEngine welche MessageQueues verwalten kann. Weder MySQL noch einer der MySQL-Banches liefert diese StorageEngine mit, so dass der Anwender sich selbst um die (einfache) Installation kümmern muss.

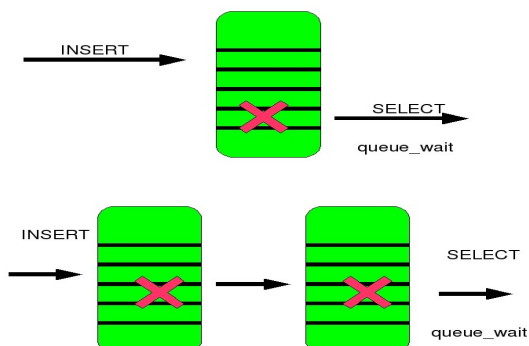


Abb. 3: Q4M sorgt dafür, dass gelesene/abgeholte Nachrichten – als übermittelnd geltend – gelöscht werden.

## Big Data

### Spider

Ein wichtiger Punkt im NoSQL-Hype ist der Zugriff auf große Datenmengen. Damit sind erst mal Datenmengen gemeint, welche nicht mehr von einem Server/Rechner alleine gehandelt werden können.

Spider erlaubt hier ein transparentes Sharding/Partitionieren – ohne dass hier disjunkte Datenmengen gefunden werden müssen. Die SpiderStorageEngine ermöglicht es beim Zugriff auf Partitionen einer Tabelle diese an andere Server weiter zu reichen. Somit sieht die Applikation zwar eine Tabelle, de facto kann die Abfrage an eine große Anzahl von Servern gehen.

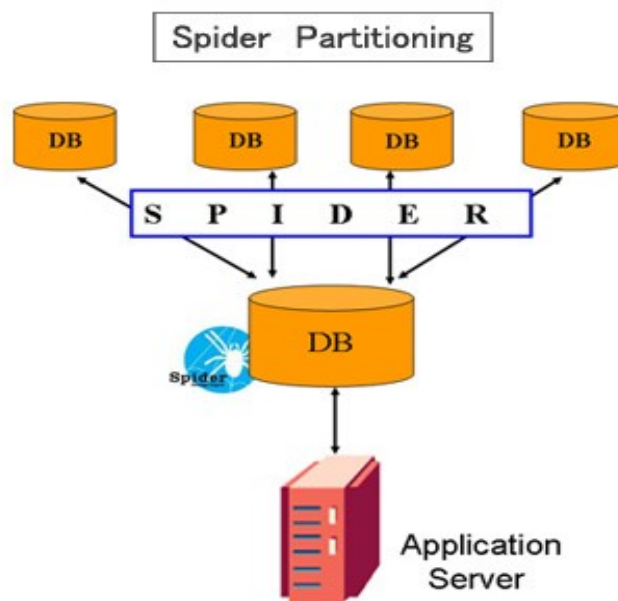


Abb. 4: Mit Spider werden Daten von anderen Server transparent aggregiert.

Bild: <http://spiderformysql.com/product.html>

Die Syntax lautet:

```
CREATE TABLE spider (id int primary key, .. ) ENGINE=SPIDER
  CONNECTION 'table "spiderc1", user "spider", password "spider"'
PARTITION BY RANGE (id)
(
  PARTITION p1 VALUES LESS THAN (10000)
    comment 'host "192.168.178.91", database "test"',
  PARTITION p2 VALUES LESS THAN (30000)
    comment 'host "192.168.178.92", database "test"',
  PARTITION p3 VALUES LESS THAN (50000)
    comment 'host "192.168.178.93", database "test2"'
);
```

## **Schlußwort**

Das Ecosystem von MySQL bietet eine große Anzahl an Lösungen welche die Einführung von neuen NoSQL-Produkten überflüssig machen (können). Die Verwendung von MySQL-'Erweiterungen' hat zudem den Charme, dass die wichtigen Daten nicht ja nach Verwendung mehrfach in der Serverlandschaft verteilt werden müssen, sondern an einem Zentralen Punkt (via SQL) verwaltet werden.

## **Kontaktadresse:**

### **Name**

Erkan Yanar

E-Mail

[erkan.yanar@linsenraum.de](mailto:erkan.yanar@linsenraum.de)

Internet:

[linsenraum.de/erkules](http://linsenraum.de/erkules)

Xing:

[https://www.xing.com/profile/Erkan\\_Yanar](https://www.xing.com/profile/Erkan_Yanar)