

Partitionieren über Rechengrenzen hinweg



Erkan Yanar

Frankfurt am Main

Schlüsselworte:

MySQL, StorageEngine, Sharding, NoSQL, Partitionierung, Federated

Einleitung

MySQL ist in vielen Bereichen modularisiert und kann (dynamisch) durch Plugins erweitert werden. Dieses Konzept/Modularität von MySQL ist eines der Alleinstellungsmerkmale dieser Datenbank. So existiert auch die StorageEngines eine API. (Zur Information bei MySQL wird die StorageEngine pro Tabelle definiert.

Kentoku Shiba hat eine StorageEngine geschrieben, welche wie eine Kombination von Partitionierung und Federated scheint.

Sprich mit der von Kentoku Shiba entwickelten StorageEngine können (Tabellen)Partitionen erstellt werden, wobei für jede Partition bestimmt werden kann, ob diese physisch auf einer anderen MySQL-Instanz zu finden ist. Grob zu vergleichen mit Database Link von Oracle. Dies erlaubt unter anderem ein transparentes besseres Sharding und (damit) den Zugriff auf sehr große Datenmengen, welche von vielen Rechnern mit deren Rechenpower prozessiert werden.

How to get

SpiderSE gehört nicht zum Standardumfang von MySQL oder einem der MySQL-Banches (MariaDB, Percona). Auf <https://launchpad.net/spiderformysql> sind sowohl der Quellcode, wie auch Binärpakete zu finden.

Einbinden der StorageEngine

SpiderSE kommt als dynamisch ladbares Plugin kann mit der Anweisung

```
INSTALL PLUGIN spider SONAME 'ha_spider.so';
```

zur Laufzeit eingebunden werden. Da Spider noch Verwaltungstabellen in mysql.* benötigt ist das einspielen eines mitgelieferten Installskriptes zu empfehlen.

```
# mysql mysql < install_spider.sql
```

Diese Skript installiert auch – wie oben angegeben – das Plugin.

Ein schneller Check:

```
mysql> show engines;
```

```
+-----+-----+-----+
+-----+-----+-----+
+-----+-----+-----+
| Engine                                     | Support | Comment
| Transactions | XA      | Savepoints |
+-----+-----+-----+
+-----+-----+-----+
| SPIDER                                           | YES      | Spider storage engine
| YES          | YES     | NO          |
| InnoDB      | DEFAULT | Supports transactions, row-level
locking, and foreign keys | YES      | YES | YES |
| MRG_MYISAM  | YES     | Collection of identical MyISAM
tables          | NO       | NO | NO |
| PERFORMANCE_SCHEMA | YES     | Performance Schema
| NO          | NO     | NO          |
| CSV         | YES     | CSV storage engine
| NO          | NO     | NO          |
| MEMORY      | YES     | Hash based, stored in memory,
useful for temporary tables | NO       | NO | NO |
| MyISAM      | YES     | MyISAM storage engine
| NO          | NO     | NO          |
+-----+-----+-----+
+-----+-----+-----+
+-----+-----+-----+
7 rows in set (0.00 sec)
```

Spider nutzen

Eine Tabelle von Typ Spider wird wie alle Tabellen mit CREATE TABLE ... ENGINE= erstellt. (Zwar ist die Syntax CREATE TABLE, jedoch ist in diesem Fall es für die Begrifflichkeit sinnvoller an ein CREATE LINK zu denken.)

Die CONNECTION und COMMENT Definition bestimmt wo die passenden realen Tabellen zu suchen sind. CONNECTION definiert die globalen Einstellungen und in CONNECTION können die Einstellungen für die jeweilige Partition lokal überschrieben (oder gesetzt) werden.

```
CREATE TABLE tbl_a(
    col_a int,
    col_b int,
    primary key(col_a)
) ENGINE = SPIDER
CONNECTION ' table "tbl_a", user "ich", password "auch" '
partition by range( col_a )
(
    PARTITION p1 VALUES LESS THAN (1000)
    COMMENT 'host "192.168.178.25", port "3306"',
    PARTITION p2 VALUES LESS THAN (2000)
```

```

COMMENT 'host "192.168.178.26", port "3306"',
PARTITION p3 VALUES LESS THAN (MAXVALUE)
COMMENT 'host "192.168.178.27", port "3306"'

```

);

Die referenzierten Tabellen können jede mögliche StorageEngine nutzen. Auch kann auf die referenzierten Tabellen ohne 'Umweg' zugegriffen werden. So ist es möglich direkt auf die entsprechende Instanz zugegriffen werden kann.

Spider ermöglicht via XA-Transactions Transaktionen auf der Tabelle, welche sich auf verschiedenen Servern befinden, als eine Transaktion durchzuführen. Dies ist mit simplen Sharding nicht machbar. Zudem läßt Spider die Anfragen vom Zielsystem bearbeiten (Condition Pushdown). Federated(X) muß sich alle Rows holen um Bedingungen zu prozessieren.

Szenarien

Sharding

Mit SpiderSE wird transparent zur Applikation ein Sharding aufgebaut. Dies erlaubt einfaches Scaleout, ohne z.B. Applikationen anpassen zu müssen.

Ein weiterer Vorteil von Sharding via SpiderSE ist, dass weiterhin eine (XA-)Transaktion über die Shards/Partitionierungen gemacht werden kann. Dies ist mit klassischem Sharding als nur disjunkte Teilmengen nicht möglich.

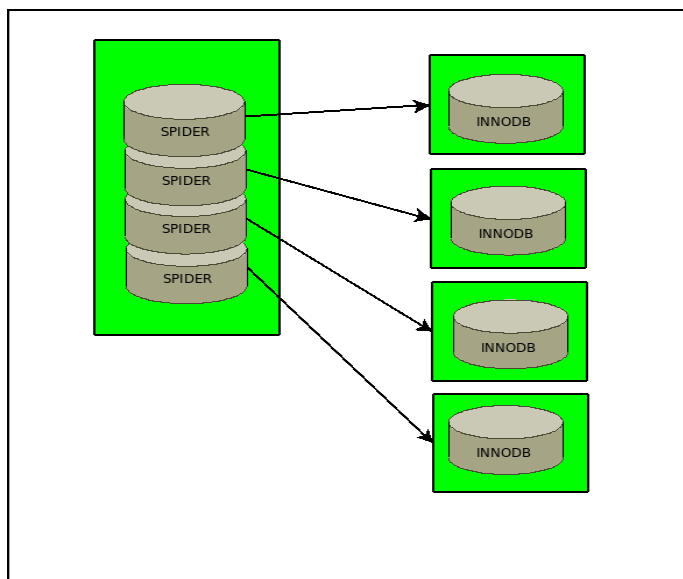


Abb. 1: Bei einem Zugriff auf die Partitionen einer Tabelle der Engine Spider werden die Zugriffe auf andere Tabellen umgeleitet.

Data Sharing

Mehrere Server können auf die selben Tabellen zugreifen und gemeinsam nutzen. Hier sind viele Szenarien möglich. Das wohl einfachste Szenario ist, das zwei MySQL Instanzen auf der selben Tabelle (eine Instanz greift via Spider drauf zu) arbeiten.

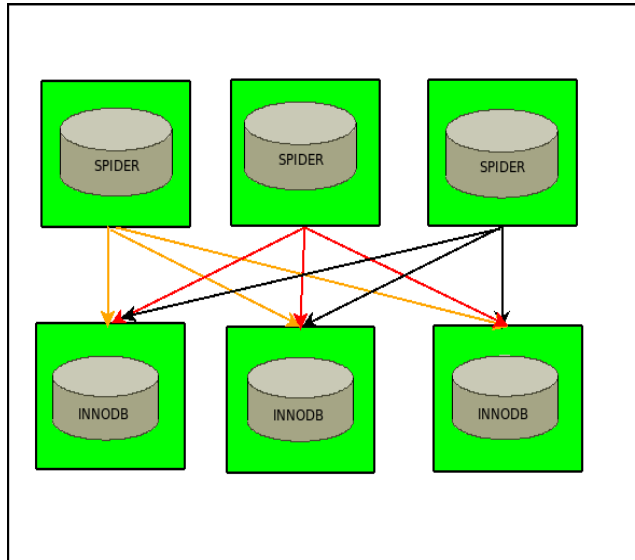


Abb. 2: Mehrere MySQL Instanzen können via Spider auf die selben Datensätze zugreifen.

Kaskadieren von Partitionen

MySQL ermöglicht zwei Level von Partitionen: Partition und Subpartition. Indem die referenzierte Tabelle selbst partitioniert werden kann sind eine Schachtelung von 4 Partitionen (Partition, Subpartition – Partition, Subpartition) möglich.

Parallele Replikation

Für die Applikation ist es eine Tabelle, da die jeweiligen Partitionen aber nun Tabellen auf separaten Rechnern sind. Wird jede Partition/Tabelle parallel/eigenständig repliziert.

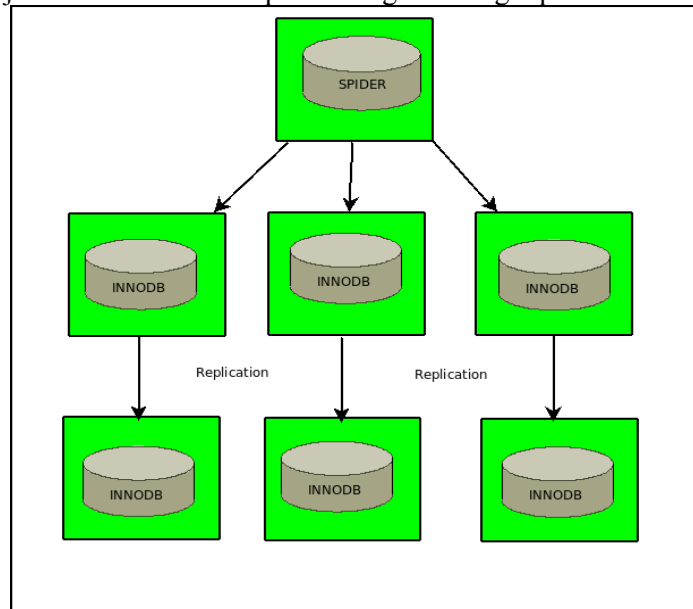


Abb. 3: Via Spider funktionieren die 3 Tabellen/Instanzen in der oberen Reihe wie eine. Diese eine Tabelle wird faktisch über drei Replikationskanäle repliziert anstatt einem.

Synchrone Replikation

Werden Änderungen einer Tabelle via Triggers in eine Tabelle vom Typ/ENGINE Spider übertragen, kommt die Anweisung erst zurück, wenn der Trigger auf der Spider-Tabelle ausgeführt ist. Das ist dann der Fall, wenn die Daten in der von Spider referenzierten Tabelle angekommen sind.

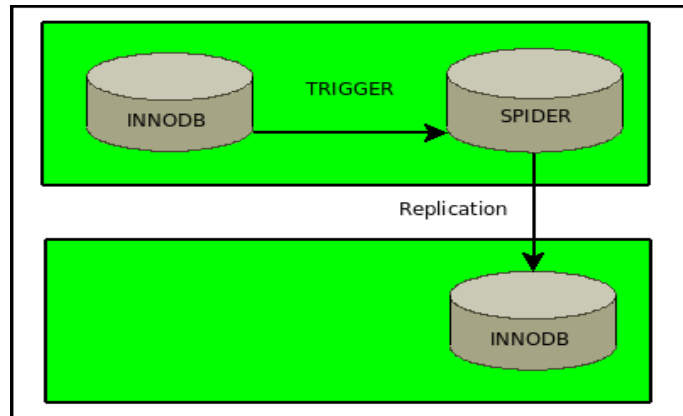


Abb. 4: Spider ermöglicht via Trigger eine synchrone Replikation.

Fazit

Spider ist ein faszinierendes Plugin, welches viel mehr Aufmerksamkeit verdient hat.

Kontaktadresse:

Name

Erkan Yanar

E-Mail

erkan.yanar@linsenraum.de

Internet:

linsenraum.de/erkules

xing:

https://www.xing.com/profile/Erkan_Yanar