

InnoDB Tablespace Cleanup mit pt-online-schema-change

Mit der Zeit wird das Datenfile (.ibd) einer InnoDB Tabelle durch das Hinzufügen und Löschen von Daten immer größer. Obwohl die Tabelle eigentlich viel kleiner ist wird der benötigte Plattenplatz nicht immer wieder freigegeben.

Die Voraussetzung für die folgenden Tipps ist, dass `innodb_file_per_table=on` gesetzt und beim PXC die `wsrep_OSU_method=T0I` eingestellt ist.

Bei Tabellen ohne große Last kann das schnell und einfach mit einem `mysqloptimize -A` behoben werden (alle Datenbanken und Tabellen werden "optimiert"). Dabei wird die Tabelle neu erstellt.

Handelt es sich um eine sehr aktive Tabelle und soll der Impact dieser Aktion nicht so groß sein (und zum Beispiel einen Percona Cluster sperren), dann kann man das Tool pt-online-schema-change aus dem Percona Toolkit nutzen. Das Tool kann die Tabellenstruktur ändern ohne Lese-/Schreibzugriffe zu blockieren. Normalerweise wird es verwendet, um z.B. ein neues Feld oder einen Index hinzuzufügen.

Ein `ALTER TABLE ENGINE=InnoDB` bewirkt hier aber das gleiche wie ein `mysqloptimize`. Ich habe mir dafür ein Script geschrieben, das alle Tabellen einer Datenbank durcharbeitet. Einziges Argument ist der Datenbankname.

```
#!/bin/bash

DB=${1}

TABLES=$(mysql -N -B ${DB} -e "show tables;")

for table in ${TABLES}; do
    echo "----- Processing ${DB} -- ${table} -----"
    pt-online-schema-change --execute --alter "ENGINE=InnoDB" --progress=time,5
    D=${DB},t=${table}
    echo
done
```

Das sieht dann so aus:

```
# ./optimize-table-pt-osc.sh my_db
----- Processing my_db -- AuthAssignment -----
No slaves found. See --recursion-method if host dbserver-01-a has slaves.
Not checking slave lag because no slaves were found and --check-slave-lag was not specified.
Operation, tries, wait:
  analyze_table, 10, 1
  copy_rows, 10, 0.25
```

```
create_triggers, 10, 1
drop_triggers, 10, 1
swap_tables, 10, 1
update_foreign_keys, 10, 1
Altering `my_db`.`AuthAssignment`...
Creating new table...
Created new table my_db._AuthAssignment_new OK.
Altering new table...
Altered `my_db`.`_AuthAssignment_new` OK.
2023-08-31T10:49:15 Creating triggers...
2023-08-31T10:49:15 Created triggers OK.
2023-08-31T10:49:15 Copying approximately 157 rows...
2023-08-31T10:49:15 Copied rows OK.
2023-08-31T10:49:15 Analyzing new table...
2023-08-31T10:49:15 Swapping tables...
2023-08-31T10:49:15 Swapped original and new tables OK.
2023-08-31T10:49:15 Dropping old table...
2023-08-31T10:49:15 Dropped old table `my_db`.`_AuthAssignment_old` OK.
2023-08-31T10:49:15 Dropping triggers...
2023-08-31T10:49:15 Dropped triggers OK.
Successfully altered `my_db`.`AuthAssignment`.
```

Bei einem Kunden konnte ich so den benötigten Platz von 450 GB auf 90 GB verringern.

Revision #2

Created 31 August 2023 08:41:02 by magenbrot

Updated 31 August 2023 10:59:46 by magenbrot