
Use Cases for Partitioning

Bill Karwin

Percona, Inc. 2011-02-16

Why “Use Cases”?

- ★ Anyone can read the reference manual:
 - ★ <http://dev.mysql.com/doc/refman/5.1/en/partitioning.html>
- ★ This talk is about *when* and *why* to use partitioning
- ★ and *what options* to use

Partitioning in MySQL 5.1

- ★ Horizontal partitioning (by rows)
- ★ Query across partitions transparently
- ★ Helps archiving old data
- ★ Query only against relevant partition of data
- ★ Supports more data than a single file or filesystem

Partitioning in MySQL 5.1

- ★ Choices for Partition type
 - ♦ RANGE
 - ♦ LIST
 - ♦ HASH
 - ♦ KEY
- ★ How to choose?

Partitioning in MySQL 5.5

- ★ New Partition types:
RANGE COLUMNS, LIST COLUMNS
 - ◆ Define ranges or lists against multiple columns
 - ◆ Prune partitions for expressions over multiple columns
 - ◆ Partition range or list against non-integer data types
DATE, DATETIME, CHAR, VARCHAR
- ★ TRUNCATE PARTITION

Partitioning Is Not

- ★ Load balancing
 - ♦ Alternative: Master-Slave Replication
 - ♦ Alternative: MySQL Proxy
 - ♦ Alternative: NDB Cluster

Partitioning Is Not

- ★ Hot backup
 - ♦ No redundancy. Table is split, not cloned.
 - ♦ Alternative: Master-Master Replication
 - ♦ Alternative: MMM (<http://mysql-mmm.org/>)

Partitioning Is Not

- ★ Software RAID
 - ♦ Alternative: Software RAID
 - ♦ Alternative: Hardware RAID

Partitioning Is Not

- ★ Parallel query

- ◆ Alternative: NDB Cluster
- ◆ Alternative: MySQL Proxy
- ◆ Alternative: Shard-Query
(<http://code.google.com/p/shard-query/>)

- ★ Parallel aggregation

- ◆ Alternative: MySQL Proxy
- ◆ Alternative: Shard-Query

Use Case: Query by PK

- ★ If you want rows to fill partitions approximately equally
- ★ If you don't care which partition a given row lives in
- ★ CREATE TABLE Tweets (
 TweetId SERIAL PRIMARY KEY,
 ...
) PARTITION BY **HASH**(TweetId) PARTITIONS 8;
- ★ SELECT * FROM Tweets WHERE TweetId = 1234;

Use Case: Query by PK

- ★ If you don't care if partitions are filled as equally
- ★ If you want to add/drop/merge/split partitions faster
- ★

```
CREATE TABLE Tweets (  
    TweetId SERIAL PRIMARY KEY,  
    ...  
) PARTITION BY LINEAR HASH( TweetId ) PARTITIONS 8;
```
- ★

```
SELECT * FROM Tweets WHERE TweetId = 1234;
```
- ★

```
ALTER TABLE Tweets COALESCE PARTITION 4; -- remove 4 partitions
```
- ★

```
ALTER TABLE Tweets ADD PARTITIONS 6; -- add 6 partitions
```

Use Case: Query by PK

- ★ If your primary key is not an integer
- ★ CREATE TABLE Tweets (
 TweetToken BINARY(16) PRIMARY KEY,
 ...
) PARTITION BY **KEY**(TweetToken) PARTITIONS 8;
- ★ SELECT * FROM Tweets WHERE TweetToken = UNHEX(MD5('1234'));

Use Case: Query by Expression

- ★ If you want rows with close values to be stored together
- ★ If you can make an integer expression to group rows
- ★

```
CREATE TABLE Tweets (  
    Tweet VARCHAR(140),  
    TweetDate DATETIME,  
    ...  
) PARTITION BY RANGE( TO_DAYS(TweetDate) )  
PARTITION p0 VALUES LESS THAN (734472), -- 2010-12-01  
PARTITION p1 VALUES LESS THAN (734503), -- 2011-01-01  
PARTITION p2 VALUES LESS THAN (734534), -- 2011-02-01  
PARTITION p3 VALUES LESS THAN (734562), -- 2011-03-01  
PARTITION p4 VALUES LESS THAN (734593); -- 2011-04-01
```

Use Case: Query by Expression

- ★ If you query with a compatible expression, MySQL touches only the relevant partition(s).
- ★

```
SELECT * FROM Tweets  
WHERE TO_DAYS(TweetDate) >= TO_DAYS('2011-02-16');
```
- ★ Partition pruning by expression supports only functions YEAR(), TO_DAYS(), and in MySQL 5.5, TO_SECONDS(). You can use other functions in expressions, but not benefit from pruning.

Use Case: INSERT

- ★ INSERT INTO Tweets (Tweet, TweetDate)
VALUES ('Percona Live SF 2/16', NOW());

Use Case: UPDATE

- ★ UPDATE Tweets SET IsDeleted = 1
WHERE TweetToken = UNHEX(MD5('1234'));
- ★ Does this benefit from partition pruning?
 - ♦ Yes, if partition on TweetToken
 - ♦ No, if partition on TweetDate
- ★ UPDATE Tweets SET IsDeleted = 1
WHERE TweetToken = UNHEX(MD5('1234'))
AND TO_DAYS(TweetDate) = TO_DAYS('2011-02-16');

Use Case: UPDATE

- ★ UPDATE Tweets SET IsDeleted = 1
WHERE TweetToken = UNHEX(MD5('1234'))
AND TO_DAYS(TweetDate) = TO_DAYS('2011-02-16');
- ★ Does this benefit from partition pruning?
 - ♦ Yes, if partition on TweetToken
 - ♦ Yes, if partition on TweetDate

Use Case: DELETE / Archiving

- ★ ALTER TABLE Tweets DROP PARTITION p0;
 - ♦ Fast, works with no temporary table
- ★ Only works with RANGE / LIST partitioning
Why?
 - ♦ DROP a RANGE / LIST partition deletes data rows in partition
 - ♦ COALESCE a HASH / KEY partition set redistributes rows, which rebuilds the table
- ★ If you want to archive / retire old data, use RANGE / LIST partition types

Use Case: Really Big Table

- ★ Larger than your filesystem
 - ♦ You can specify physical path for each partition:
 - ♦ CREATE TABLE Tweets (. . .)
PARTITION BY RANGE(TO_DAYS(TweetDate))
PARTITION p0 VALUES LESS THAN (734472) (
DATA DIRECTORY = '/disk0/data',
INDEX DIRECTORY = '/disk0/index'
),
PARTITION p1 VALUES LESS THAN (734503) (
DATA DIRECTORY = '/disk1/data',
INDEX DIRECTORY = '/disk1/index'
),
. . .

-
- ★ We'll be at the MySQL Conference April 11-14, 2011. Please join us!
 - ★ Register with 20% discount using this code:

mys11pka