

后盾网 人人做后盾

www.houdunwang.com

MySQL表分区

后盾网 2011-2016

当Mysql数据量达到千万级进，并发量大时，即时简单的查询Mysql响应是不尽如人意。

可以能过以下方式处理

1. 更换性能更好的硬件，加大内存
2. 增加Mysql查询缓存与key缓存
3. 使用第3方如memcache等nosql技术
4. 数据库进行分区处理，减少表大小

Mysql性能优化方案

通俗地讲表分区是将一大表，根据条件分割成若干个小表。
mysql5.1开始支持数据表分区了。

某用户表的记录超过了600万条，那么就可以根据入库日期将表分区，也可以根据所在地将表分区。当然也可根据其他的条件分区

Mysql表分区

1. 与单个磁盘或文件系统分区相比，可以存储更多的数据。
2. 对于那些已经失去保存意义的数据，通常可以通过删除与那些数据有关的分区，很容易地删除那些数据。相反地，在某些情况下，添加新数据的过程又可以通过为那些新数据专门增加一个新的分区，来很方便地实现。
3. 一些查询可以得到极大的优化，这主要是借助于满足一个给定WHERE语句的数据可以只保存在一个或多个分区内，这样在查找时就不用查找其他剩余的分区。因为分区可以在创建了分区表后进行修改，所以在第一次配置分区方案时还不曾这么做时，可以重新组织数据，来提高那些常用查询的效率。
4. 涉及到例如SUM()和COUNT()这样聚合函数的查询，可以很容易地进行并行处理。这种查询的一个简单例子如“SELECT salesperson_id, COUNT (orders) as order_total FROM sales GROUP BY salesperson_id;”。通过“并行”，这意味着该查询可以在每个分区上同时进行，最终结果只需通过总计所有分区得到的结果。

Mysql分区特点

查看插件有无partition，这是决定能否实现分区的基本条件

- show plugins;

查看Mysql是否支持分区

RANGE 分区：基于属于一个给定连续区间的列值，把多行分配给分区。

这些区间要连续且不能相互重叠，使用VALUES LESS THAN或VALUES LESS THAN MAXVALUE操作符来进行定义

Range分区

```
drop database if exists demo;
create database demo charset utf8;
use demo;
create table partition_table(id int,user varchar(30))
partition BY RANGE (id) (
    partition p0 VALUES LESS THAN (1000),
    partition p1 VALUES LESS THAN (2000),
    partition p2 VALUES LESS THAN (3000),
    partition p3 VALUES LESS THAN MAXVALUE
);
\d $$
create procedure insert_data()
begin
declare i int default 5000;
while i>0 do
    insert into partition_table(id,user)values(i,left(md5(i),5));
    set i=i-1;
end while;
end;
$$
\d;
call insert_data();
```

Range分区 (数值)

```
drop database if exists demo;
create database demo charset utf8;
use demo;
create table test(id int,name varchar(100),addtime date)
partition BY RANGE (year(addtime)) (
    partition p0 VALUES LESS THAN (2000),
    partition p1 VALUES LESS THAN (2010),
    partition p3 VALUES LESS THAN MAXVALUE
);
\q
create procedure insert_data()
begin
declare i int default 10000;
declare y date;
while i>0 do
    insert into test(id,name,addtime)values(i,md5(i),date_add("1990-01-01",interval +i day));
    set i=i-1;
end while;
end;
\q
call insert_data();

explain select * from test where year(addtime)=2001 //查看查询条数
```

Range分区(时间)

1. 当需要删除一个分区上的“旧的”数据时,只删除分区即可。如果你使用上面最近的那个例子给出的分区方案,你只需简单地使用“ALTER TABLE employees DROP PARTITION p0;”来删除所有在1991年前就已经停止工作的雇员相对应的所有行。对于有大量行的表,这比运行一个如“DELETE FROM employees WHERE YEAR (separated) <= 1990;”这样的一个DELETE查询要有效得多。
2. 经常运行直接依赖于用于分割表的列的查询。例如,当执行一个如“SELECT COUNT(*) FROM employees WHERE YEAR(separated) = 2000 GROUP BY store_id;”这样的查询时,MySQL可以很迅速地确定只有分区p2需要扫描,这是因为余下的分区不可能包含有符合该WHERE子句的任何记录

Range分区

LIST 分区：类似于按RANGE分区，区别在于LIST分区是基于列值匹配一个离散值集中的某个值来进行选择。

比如我们可以按照部门ID来进行分区，这样当删除部门数据时只要删除此部门分区就可以了。

如果试图插入列值（或分区表达式的返回值）不在分区值列表中的一行时，那么“INSERT”查询将失败并报错。例如，下面的表中如果我们插入一个数据但没有部门时将报错。

LIST分区没有类似如“VALUES LESS THAN MAXVALUE”这样的包含其他值在内的定义。将要匹配的任何值都必须在值列表中找到。

LIST分区

```
drop database if exists demo;
create database demo charset utf8;
use demo;
create table part(cid int,cname varchar(100))
partition BY LIST (cid) (
    partition p1_2_3 VALUES IN (1,2,3),
    partition p4_5 VALUES IN (4,5,6),
    partition p6_7 VALUES IN (7,8),
    partition p8_9 VALUES IN (9,10)
);
\d $$
```

List分区

HASH分区主要用来确保数据在预先确定数目的分区中平均分布。在RANGE和LIST分区中，必须明确指定一个给定的列值或列值集合应该保存在哪个分区中；而在HASH分区中，MySQL 自动完成这些工作，你所要做的只是基于将要被哈希的列值指定一个列值或表达式(只要它们返回一个既非常数、也非随机数的整数)，以及指定被分区的表将要被分割成的分区数量。

要使用HASH分区来分割一个表，要在CREATE TABLE 语句上添加一个“PARTITION BY HASH (expr)”子句，其中“expr”是一个返回一个整数的表达式。它可以仅仅是字段类型为MySQL 整型的一列的名字。你可以在后面添加 “PARTITIONS num”子句，其中num 是一个非负的整数，它表示表将要被分割成分区的数量，如果不设置为默认分1个区。

每当插入或更新（或者可能删除）一行，这个表达式都要计算一次；这意味着非常复杂的表达式可能会引起性能问题，尤其是在执行同时影响大量行的运算（例如批量插入）的时候。

HASH分区

```
drop database if exists demo;
create database demo charset utf8;
use demo;
create table test(id int,name varchar(100),addtime date)
partition BY HASH (YEAR(addtime))
PARTITIONS 10;
\d $$
create procedure insert_data()
begin
declare i int default 100000;
declare y date;
while i>0 do
    insert into test(id,name,addtime)values(i,md5(i),date_add("1900-01-01",interval +i day));
    set i=i-1;
end while;
end;
$$
\d ;
call insert_data();
```

HASH分区

按照KEY进行分区类似于按照HASH分区，除了HASH分区使用的用户定义的表达式，而KEY分区的 哈希函数是由MySQL 服务器提供。

Key分区

```
drop database if exists demo;
create database demo charset utf8;
use demo;
create table test(id int,name varchar(100),addtime date)engine myisam charset utf8
partition BY KEY (addtime)
PARTITIONS 10;
\q $$
create procedure insert_data()
begin
declare i int default 100000;
declare y date;
while i>0 do
    insert into test(id,name,addtime)values(i,md5(i),date_add("1900-01-01",interval +i day));
    set i=i-1;
end while;
end;
$$
\q ;
call insert_data();
```

Key分区