Mysql数据分区设计方案

# 问题描述

app\_car\_gps\_log,app\_car\_obd\_log两张表的每天的日志量比列3：1 ,日均每台车机的日志量大概是1000条左右。

mysql维护一个索引节点需要24个字节

app\_car\_gps\_log单条记录所占的字节总数为：

1、数据占用空间 78 字节

2、索引占用空间 24+(24+8+8+8) +(24+8+8+8) =120 字节

3、数据+索引占用空间 198字节

4、单台车机每天占用空间198\*750=148500 字节

app\_car\_obd\_log单行所占记录128个字节

1、数据占用空间 128字节

2、索引占用空间24+(24+8+8+8) +(24+8+8+8) =120字节

3、数据+索引占用空间248字节

4、单台车机每天占用空间248\*250=62000 字节

结论：

1、单台车机两张记录每天占用空间为148500+62000=210500

2、10万车机两张表记录每天占用空间为210500\*10^5/2^30=20G

# 解决方案

一、mysql表分区设计

1、设计单个schema的储存能力为存储总能力10亿条

2、设计单表单分区的设计为10^7条数据 占用空间约为2G

3、设计单表分区总数为128个分区，占用空间 256G。

4、结论单个shcema 可以储存10万台车机10天的数据

二、cobar 分区设计

1、一个分区设计为三个schema，存储能力为30亿条

2、10万车机一个月的数据量，空间占用约为600G左右

3、cobar路由策略 ：

根据time值查询分区容量表确定所在的分区，然后通过hash(car\_id)确定所在的schema;

4、当当前分区的空间占用超过90%，开始扩容，初始化下一个分区。

5、分区表和Schema的对应关系

dbcluster\_ part\_schema

|  |  |  |  |
| --- | --- | --- | --- |
| Id | part\_code | shcema\_id | capacity |
| 1 | Part1 | Dbtest1 | 10^9 |
| 2 | Part1 | Dbtest2 | 10^9 |
| 3 | Part1 | Dbtest3 | 10^9 |
| 4 | Part2 | Dbtest4 | 10^9 |
| 5 | Part2 | Dbtest5 | 10^9 |
| 5 | Part2 | Dbtest6 | 10^9 |

分区容量统计表

dbcluter\_part\_capactiy

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | Part\_code | Start\_date | End\_date | total\_capacity | current\_capacity |
| 1 | Part1 | 2014-05-01 | 2014-05-30 | 10^12 | 10^12 |
| 2 | part2 | 2014-06-01 | 2014-06-30 | 10^12 | 10^9 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

容量统计策略:

1. 单schema的两张表id，采用自增长策略,
2. 每个Id的初始值为10^10\*(shema\_id),然后在此基础上增长
3. (select max(id) from app\_car\_gps\_log-该表的Id初始值)

+ (select max(id) from app\_car\_obd\_log-该表的Id初始值)

c) 总容量为三个shcmea的两张表记录的和

6、10万车机一年的schema的总数为36个。

7、关于如果每天日志的生成量大于mysql的插入速度的话，需要扩容schema。

，如果扩容的话，会导致新的shcema没有被利用 ，这样就需要加新的分区。然后把新的分区，然后让一个分区做数据均衡。待数据均衡工作完成后，然后在修改分区容量表，然后该分区能压入新的时间段数据。增加新的时间段。这样该分区可能有多个时间段组成。

8、关于增加分区的时候，怎么保证不停服务问题。

如果发现容量接近70%-80%的时候，就加入新的分区。当时间到达某个点后，自动把数据压入新的分区。

9、下面为两张表的表结构优化过后的数据.

优化表结构遵循的原则:

a)所有的精度型数据，统一用int类型表示。

b)date,timestamp,统一用int类型表示。

c)car\_id把以前的uuid转为为int类型存储。

表结构 app\_car\_gps\_log

表里double,date,times ,timestamp统一用int 储存。

分区做hash分区根据car\_id

主键用id,car\_id复合主键

这个因为要用car\_id分区。

表里需要维护两个索引分别为主键索引(id,car\_id)、复合索引(car\_id,time )、复合索引(din,time)

CREATE TABLE `app\_car\_gps\_log` (

`id` bigint(20) unsigned NOT NULL auto\_increment ,

`din` varchar(20) DEFAULT NULL,

`car\_id` int(11) NOT NULL DEFAULT '0',

`time` int(11) DEFAULT NULL,

`insert\_timestamp` int(11) DEFAULT NULL,

`receive\_time` int(11) DEFAULT NULL,

`longitude` int(11) DEFAULT NULL,

`latitude` int(11) DEFAULT NULL,

`altitude` int(11) DEFAULT NULL,

`speed` int(11) DEFAULT NULL,

`heading` int(11) DEFAULT NULL,

`battery\_voltage` int(11) DEFAULT NULL,

`kmph` int(11) DEFAULT NULL,

`satellites` smallint(6) DEFAULT NULL,

`acc` smallint(6) DEFAULT NULL,

`valid` tinyint(1) DEFAULT NULL,

`code` tinyint(1) DEFAULT '0',

`hide` tinyint(1) DEFAULT '0',

PRIMARY KEY (`id`,`car\_id`),

KEY `idx\_car\_gps\_log\_car\_id` (`car\_id`,`time`),

KEY `idx\_car\_gps\_log\_din` (`din`,`time`)

) ENGINE=InnoDB DEFAULT CHARSET= utf8

PARTITION BY HASH (car\_id)

PARTITIONS 128

表结构 app\_car\_obd\_log

表里double,date,times ,timestamp统一用int 储存。

分区做hash分区根据car\_id

主键用id,car\_id复合主键

这个因为要用car\_id分区。

CREATE TABLE `app\_car\_obd\_log` (

`id` bigint(20) unsigned NOT NULL auto\_increment ,

`din` varchar(20) DEFAULT NULL,

`car\_id` int(11) DEFAULT NULL,

`air\_flow` int(11) DEFAULT NULL,

`tvp` int(11) DEFAULT NULL,

`pedal\_position` int(11) DEFAULT NULL,

`engine\_runtime` int(11) DEFAULT NULL,

`trouble\_mileage` int(11) DEFAULT NULL,

`engine\_payload` int(11) DEFAULT NULL,

`lfuel\_trim` int(11) DEFAULT NULL,

`ciaa` int(11) DEFAULT NULL,

`device\_runtime` int(11) DEFAULT NULL,

`insert\_timestamp` int(11) DEFAULT NULL,

`time` int(11) DEFAULT NULL,

`receive\_time` int(11) DEFAULT NULL,

`speed` int(11) DEFAULT NULL,

`rpm` int(11) DEFAULT NULL,

`mileage` int(11) DEFAULT NULL,

`bus\_mileage` int(11) DEFAULT NULL,

`fuel` int(11) DEFAULT NULL,

`residual\_fuel` int(11) DEFAULT NULL,

`per\_residual\_fuel` int(11) DEFAULT NULL,

`battery\_voltage` int(11) DEFAULT NULL,

`trouble\_light\_sin` smallint(6) DEFAULT NULL,

`malfunction\_num` smallint(6) DEFAULT NULL,

`onflow\_ct` smallint(6) DEFAULT NULL,

`coolant\_ct` smallint(6) DEFAULT NULL,

`environment\_ct` smallint(6) DEFAULT NULL,

`imap` smallint(6) DEFAULT NULL,

`fuel\_pressure` smallint(6) DEFAULT NULL,

`air\_pressure` smallint(6) DEFAULT NULL,

PRIMARY KEY (`id`,`car\_id`),

KEY `idx\_car\_obd\_log\_car\_id` (`car\_id`,`time`),

KEY `idx\_car\_obd\_log\_din` (`din`,`time`)

) ENGINE=InnoDB DEFAULT CHARSET= utf8

PARTITION BY HASH (car\_id)

PARTITIONS 128

10、pdm文件



11、cobar的使用约束

使用jdbc时，使用mysql-jdbc-5.1以上版本，推荐5.1.6或者5.1.12版本。

不支持rewriteBatchedStatements=true参数设置。默认为false

不支持useServerPrepStmts=true参数设置。默认为false

不支持跨库(数据节点)的关联操作：join、分页、排序、子查询。

BLOB, BINARY, VARBINARY字段不能使用。若特殊需求需要这三种字段，禁止使用PreparedStatement的setBlob()或setBinaryStream()方法设置参数。

不支持SAVEPOINT操作。

不支持SET语句的执行，事务和字符集设置语句除外。

对于拆分表（分库表），不能更新已有记录的拆分字段（分库字段）值。

只支持MySQL数据节点。

对于拆分表，插入操作须给出列名，必须包含拆分字段。

12、集群结构图

Cobra\_manager

client

Part3

Part2

Part1

db

db

db

db

db

db

db

db

db

db

db

Cobra\_Server

Cobra\_Server

Cobra\_Server

Cobra\_Server