

# TiDB 中的自增主键有哪些使用限制，应该如何避免？

威哥爱编程2024-09-20127 阅读6分钟专栏：V哥原创技术栈

智能总结

复制重新生成

文章主要介绍了 TiDB 中自增主键的使用限制，包括定义位置、数据类型、默认属性等，还阐述了避免这些限制的方法，如使用 AUTO\_RANDOM、SHARD\_ROW\_ID\_BITS 等，并分别通过业务场景案例详细说明了其操作步骤和注意事项。

关联问题: 如何监控TiDB主键性能UUID用于主键的弊端Snowflake算法怎样应用

基于该文章内容继续向AI提问

大家好，我是V 哥，在TiDB中使用自增主键时，确实存在一些限制和潜在的热点问题，今天的文章来聊一聊TiDB中的自增主键要怎么做。

以下是一些使用限制和如何避免它们的方法：

1. 自增主键的限制：
- 必须在主键或唯一索引列上定义。

只能定义在整型、FLOAT或DOUBLE类型的列上。

自增列不支持DEFAULT定义。

不支持使用ALTER TABLE增加AUTO\_INCREMENT属性。

默认不允许移除AUTO\_INCREMENT属性，可以通过 @tidb\_allow\_remove\_auto\_inc 来控制是否允许删除自增属性。

可以保证自增不唯一，无法保证顺序。
2. 避免自增主键限制的方法：
- 使用 AUTO\_RANDOM：TiDB提供了 AUTO\_RANDOM 属性，可以在建表时替代 AUTO\_INCREMENT 使用，这样TiDB会生成随机分布的ID，从而避免写入热点问题。

使用 SHARD\_ROW\_ID\_BITS：对于非聚簇索引主键或没有主键的表，TiDB会使用一个隐式的自增RowID。通过设置 SHARD\_ROW\_ID\_BITS，可以把RowID打散写入多个不同的Region，缓解写入热点问题。

使用分布式ID生成器：例如Snowflake算法，可以在应用层生成唯一的ID，避免依赖数据库的自增主键。

使用UUID：可以在应用层生成UUID作为主键，但需要注意UUID的性能影响。

使用Sequence序列：TiDB支持Sequence序列，可以在创建表时定义Sequence，然后使用Sequence来生成唯一的ID。

## 使用 AUTO\_RANDOM

在TiDB中， AUTO\_RANDOM 是用于解决自增主键热点问题的一种方法。以下是一个具体的业务场景案例和操作步骤：

业务场景：假设你有一个高并发的在线服务，需要为每个服务实例生成一个唯一的标识符。如果使用传统的 AUTO\_INCREMENT 自增主键，大量的写入操作可能会导致写入热点，因为所有的写入都会尝试在最后一个Region上进行，从而影响性能。

解决方案：

1. 创建表时使用 AUTO\_RANDOM：在创建表时，将主键列设置为 AUTO\_RANDOM。例如，你可以执行以下SQL语句来创建一个新表：

sql

代码解读复制代码

```
1 CREATE TABLE service_instances (  
2     id BIGINT PRIMARY KEY AUTO_RANDOM,  
3     instance_name VARCHAR(255),  
4     created_at TIMESTAMP  
5 );
```

这样，每当插入新行而没有指定 id 值时，TiDB会自动生成一个随机的 id 值。

2. 插入数据：当插入新服务实例时，不需要手动指定 id 值，TiDB会自动为每个实例生成一个唯一的 id：

sql

代码解读复制代码

```
1 INSERT INTO service_instances (instance_name, created_at) VALUES ('ServiceInstanceName', NOW());
```



威哥爱编程LV5  
华为HDE，公众号：威哥爱编程  
榜上有名 优秀作者

284  
文章

144k  
阅读

358  
粉丝

目录收起

使用 AUTO\_RANDOM

使用 SHARD\_ROW\_ID\_BITS

最后

精选内容

Pandas高级数据处理：管道操作  
Jimaks · 10阅读 · 0点赞

DrawDB：超好用的，免费数据库设计工...  
我码玄黄 · 55阅读 · 0点赞

Java使用FFM API调用SDL  
Enaium · 24阅读 · 0点赞

Qt监控系统辅屏预览/可以同时打开4个...  
feiyangqingyun · 15阅读 · 0点赞

1Panel应用推荐：Halo开源建站工具  
FIT2CLOUD飞致云 · 18阅读 · 2点赞

找对属于你的技术圈子

回复「进群」加入官方微信群



Captured by FireShot Pro: 26 1月 2025, 13:22:34

https://getfireshot.com



这将利用 `AUTO_RANDOM` 属性生成一个随机的 `id`，从而避免写入热点。

3. 获取最后插入的ID： 如果你需要获取最后插入的 `id`，可以使用 `2524665` 函数：

sql

1

SELECT 2524665;

这将返回最近一次由TiDB隐式分配的 `AUTO_RANDOM` 值。

咱们可以使用TiDB的监控工具，如Grafana，监控 `AUTO_RANDOM` 字段的性能。注意观察是否有任何写入热点的迹象，如某个TiKV节点的负载明显高于其他节点。根据监控结果调整策略。

需要注意的是：

- 不要显式地为 `AUTO_RANDOM` 字段插入值，除非打开了 `@allow_auto_random_explicit_insert` 系统变量，并且你知道你在做什么。错误的显式赋值可能会导致值耗尽。
- `AUTO_RANDOM` 只能用于 `BIGINT` 类型的主键列，并且不支持与 `AUTO_INCREMENT` 同时使用。

## 使用 SHARD\_ROW\_ID\_BITS

业务场景案例：

假设你运营一个电商平台，需要处理大量的订单数据。每个订单都需要一个唯一的订单号，而且订单数据写入数据库时必须均匀分布，以避免写入热点。如果使用自增主键，大量的写入操作可能会集中在单个TiKV节点上，导致写入热点问题。

解决方案：

1. 创建非聚簇索引表：

创建订单表时，不使用自增主键，而是使用 `SHARD_ROW_ID_BITS` 来打散行ID，从而避免写入热点。

sql

1

CREATE TABLE orders (

2

order\_id INT PRIMARY KEY NONCLUSTERED,

3

product\_id INT,

4

quantity INT,

5

created\_at TIMESTAMP,

6

INDEX product\_idx (product\_id)

7

) SHARD\_ROW\_ID\_BITS = 4 PRE\_SPLIT\_REGIONS=3;

这里 `SHARD_ROW_ID_BITS = 4` 表示行ID会被打散到16个分片中，`PRE_SPLIT_REGIONS=3` 表示在表创建后会预先切分为8个Region。

2. 插入数据： 当创建订单时，不需要手动指定 `order_id`，TiDB会自动为每个订单分配一个唯一的行ID。

sql

1

INSERT INTO orders (product\_id, quantity, created\_at)

2

VALUES (101, 2, NOW());

同样，我们使用Grafana，监控 `orders` 表的性能。注意观察是否有任何写入热点的迹象，如某个TiKV节点的负载明显高于其他节点。根据监控结果调整 `SHARD_ROW_ID_BITS` 的值。

需要注意的是：

- 不要显式地为行ID插入值，让TiDB自动分配。
- 根据业务并发度来设置合适的 `SHARD_ROW_ID_BITS` 值，以尽量解决热点Region无法打散的问题。

问题来了，如何根据业务增长调整 'SHARD\_ROW\_ID\_BITS' 的值以优化数据库性能？

在TiDB中，`SHARD_ROW_ID_BITS` 是一个表属性，用于设置隐式 `_tidb_rowid` 分片数量的bit位数，以此来解决写入热点问题。这个属性可以在创建表时指定，也可以用来修改现有表的行为。

调整 `SHARD_ROW_ID_BITS` 值的步骤：

1. 评估当前业务量和增长趋势：

- 监控当前业务对数据库的写入模式和数据增长速度。使用Grafana监控TiDB的写入流量和热点情况。

2. 决定 `SHARD_ROW_ID_BITS` 的值：

- 根据业务增长预测和当前写入模式，决定一个合适的 `SHARD_ROW_ID_BITS` 值。例如，如果当前设置为4（分成16个分片），但写入压力仍然很高，可以考虑增加该值。

3. 修改表属性：



sql

代码解读复制代码

1

ALTER TABLE your\_table\_name SHARD\_ROW\_ID\_BITS = new\_value;

sql

代码解读复制代码

1

CREATE TABLE t (a int, b int) SHARD\_ROW\_ID\_BITS = 4 PRE\_SPLIT\_REGIONS=3;

sql

代码解读复制代码

1

CREATE TABLE t (a int, b int) SHARD\_ROW\_ID\_BITS = 4 PRE\_SPLIT\_REGIONS=3;

使用 ALTER TABLE 语句来修改 SHARD\_ROW\_ID\_BITS 值：

其中 new\_value 是你基于业务增长评估后决定的新值。

4. 预切分区域：

为了确保数据能够均匀分布，可以使用 PRE\_SPLIT\_REGIONS 选项在建表后预先切分出一定数量的 Region：

这里的 PRE\_SPLIT\_REGIONS=3 表示建完表后提前切分出8个Region。

5. 监控调整后的效果：

调整 SHARD\_ROW\_ID\_BITS 后，继续使用Grafana监控系统性能，特别注意写入流量的分布情况。

6. 根据需要进一步调整：

如果写入热点仍然存在，可能需要进一步增加 SHARD\_ROW\_ID\_BITS 的值，或者考虑其他优化措施。

需要注意的是：

增加 SHARD\_ROW\_ID\_BITS 的值会增加Region的分裂数量，可能会对TiDB集群的性能和资源使用产生影响。

调整 SHARD\_ROW\_ID\_BITS 值时，需要确保PD调度器不会因为小Region合并策略而将Region重新合并。

最后

关于使用分布式ID生成器，比如雪花算法，还有使用UUID和使用Sequence序列，这里V 哥就不再介绍了，与 MySQL 一致。因此，在 TiDB 中提供了AUTO\_RANDOM 和 SHARD\_ROW\_ID\_BITS来解决热点问题，好用的很。

标签：

数据库

分布式

TiDB

话题：

创作者训练营

本文收录于以下专栏



V哥原创技术栈

专栏目录

本专栏收集 V 哥后端开发高阶内容，包括设计模式、源码剖析、算法与数据结构、高...

49 订阅 · 278 篇文章

订阅

上一篇

除了递归算法，要如何优化实现文件...

下一篇

四款负载均衡工具Nginx、HAProxy...

评论 0

登录 / 注册

即可发布评论！



暂无评论数据

Captured by FireShot Pro: 26 1月 2025, 13:22:34

https://getfireshot.com



为你推荐

对比传统数据库，TiDB 强在哪？谈谈 TiDB 的适应场景和产品能力

威哥爱编程 | 7月前 | 👁 397 | 👍 2 | 💬 评论

数据库TiDBMySQL

被追着问UUID和自增ID做主键哪个好，为什么？

码上遇见你 | 8月前 | 👁 1.4k | 👍 14 | 💬 1

后端面试MySQL

如何在 TiDB 上高效运行序列号生成服务

PingCAP | 3年前 | 👁 346 | 👍 点赞 | 💬 评论

数据库

一篇文章彻底搞懂 TiDB 集群各种容量计算方式

PingCAP | 11月前 | 👁 529 | 👍 点赞 | 💬 评论

数据库

用单库自增键来生成id了，后期怎么分库？哎，这个坑大！

58沈剑\_架构师之路 | 4年前 | 👁 2.2k | 👍 8 | 💬 3

数据库MySQL

TiDB 5.0 跨中心部署能力初探 | Joint Consensus 助力 TiDB 5.0 无畏调度

PingCAP | 3年前 | 👁 692 | 👍 2 | 💬 评论

数据库后端

TiDB 6.0 的「元功能」：Placement Rules in SQL 是什么？

PingCAP | 2年前 | 👁 204 | 👍 点赞 | 💬 评论

后端

读TiDB源码聊设计：引子

泊浮目 | 12月前 | 👁 377 | 👍 4 | 💬 2

数据库

通过 Prometheus 编写 TiDB 巡检脚本（脚本已开源，内附链接）

PingCAP | 11月前 | 👁 557 | 👍 1 | 💬 1

数据库TiDB

分库分表的 9种分布式主键ID 生成方案，挺全乎的

程序员小富 | 4年前 | 👁 5.1k | 👍 38 | 💬 5

Java分布式

TiDB 6.0 Placement Rules In SQL 使用实践

PingCAP | 2年前 | 👁 707 | 👍 2 | 💬 评论

数据库

TiDB 5.0 跨中心部署能力初探 | 中心化还是去中心化？揭秘 TiDB 5.0 事务分布式授时模块

PingCAP | 3年前 | 👁 357 | 👍 2 | 💬 评论

数据库

ShardingSphere 分库分表--第（1）篇

xiaohezi | 3年前 | 👁 2.9k | 👍 2 | 💬 评论

数据库后端

关于建表字段是否该使用not null这个问题你怎么看？

威哥爱编程 | 4月前 | 👁 146 | 👍 1 | 💬 1

数据库MySQLOracle

MySQL实战45讲\_39 | 自增主键为什么不是连续的？

财大小陈 | 2年前 | 👁 4.0k | 👍 3 | 💬 评论

MySQL