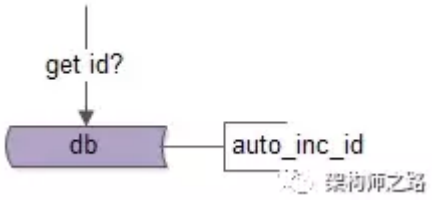
浅谈CAS在分布式ID生成方案上的应用 | 架构师之路

原创：58沈剑 架构师之路 2017-06-17

近几篇文章聊CAS被骂得较多，今天还是聊CAS，谈谈CAS在一种“分布式ID生成方案”上的应用。

所谓“分布式ID生成方案”，是指在分布式环境下，生成全局唯一ID的方法。

可以利用**DB**自增键**(auto inc id)**来生成全局唯一**ID**，插入一条记录，生成一个ID：



这个方案利用了数据库的单点特性，其优点为：

- 无需写额外代码
- 全局唯一
- 绝对递增
- 递增ID的步长确定

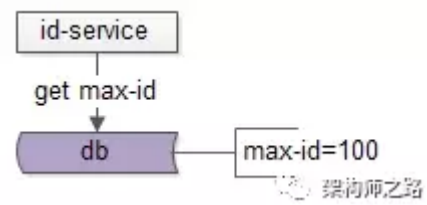
其不足为：

- 需要做数据库HA，保证生成ID的高可用
- 数据库中记录数较多
- 生成**ID**的性能，取决于数据库插入性能

优化方案为：

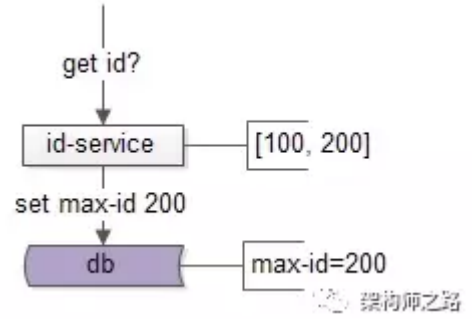
- 利用双主保证高可用
- 定期删除数据
- 增加一层服务，采用批量生成的方式降低数据库的写压力，提升整体性能

增加服务后，DB中只需保存当前最大的ID即可，在服务启动初始化的过程中，首先拉取当前的max-id：




```
select max_id from T;
```

然后批量获取一批ID，放到id-servcie内存里，并将max-id写回数据库：



update T set max_id=200;

这样，id-service就拿到了[100, 200]这一批**ID**，上游在获取ID时，不用每次都插入数据库，而是分配完100个ID后，再修改max-id的值，这样分配ID的整体性能就增加了100倍。

这个方案的优点：

- 数据库只保存一条记录
- 性能极大增强

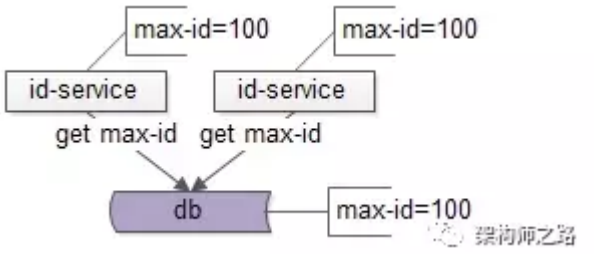
其不足为：

- 如果id-service重启，可能内存会有一段已经申请的ID没有分配出去，导致**ID**空洞，当然，这不是一个严重的问题
- 服务没有做HA，无法保证高可用

优化方案为：

- 冗余服务，做集群保证高可用

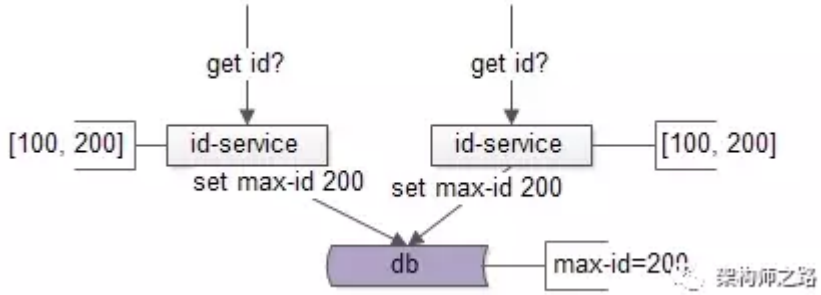
冗余了服务后，多个服务在启动过程中，进行ID批量申请时，可能由于并发导致数据不一致：



select max_id from T;

如上图所示，两个id-service在启动的过程中，同时拿到了max-id为100。

两个id-service同时对数据库的max-id进行写回：



update T set max_id=200;

写回max-id成功后，这两个**id-service**都以为自己拿到了**[100,200]**这一批**ID**，导致集群会生成重复的ID。

问题发生的原因，是并发写回时，没有对**max-id**的初始值进行比对：

id-service1写回max-id=200成功的条件是，max-id必须等于100

id-service2写回max-id=200成功的条件是，max-id也必须等于100

id-service1写回时，max-id是100，理应写回成功

id-service2写回时，max-id已经被改成了200，不应该写回成功

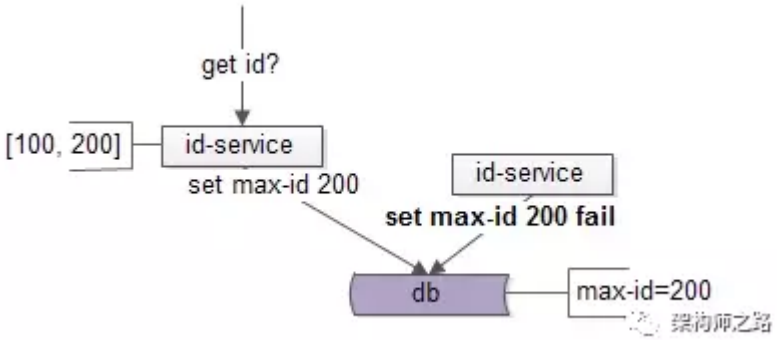
只要实施CAS乐观锁，在写回时对max-id的初始条件进行比对，就能避免数据的不一致，写回SQL由：

update T set max_id=200;

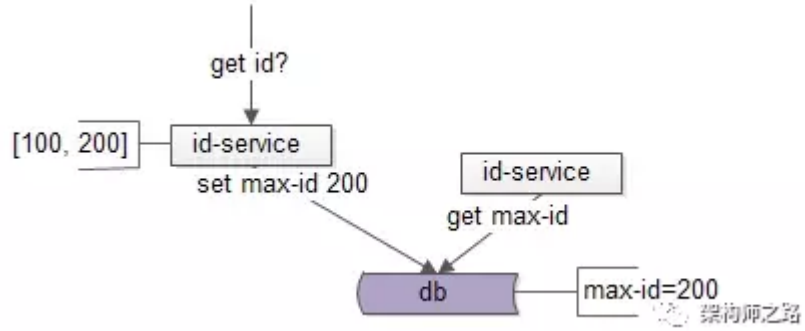
升级为：

update T set max_id=200 where max_id=100;

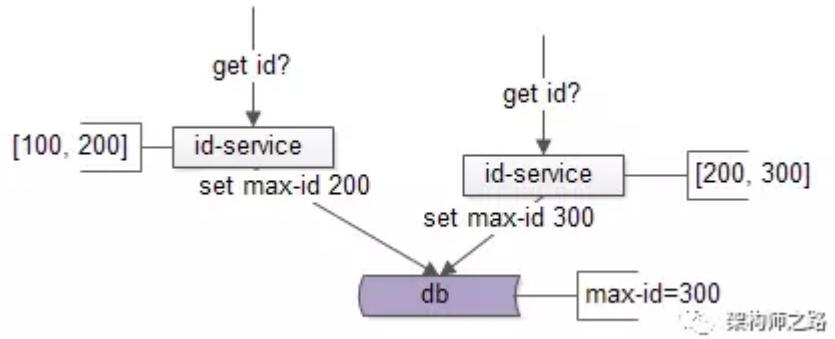
这样，id-service2写回时，就会失败：



失败后，id-service2要再次查询max-id：



此时max-id已经变为200，于是id-service2获取到了[200, 300]这一批ID，并将max-id=300写回：



update t set max_id=300 where max_id=200;

写回成功。

这种方案的好处是：

- 能够通过水平扩展的方式，达到分布式ID生成服务的无限性能
- 使用CAS简洁的保证不会生成重复的ID

其不足为：

- 由于有多个service，生成的ID 不是绝对递增的，而是趋势递增的

本文介绍了CAS在分布式ID生成方案上的一种应用，更多的分布式ID生成方案，请参考《细聊分布式ID生成器架构》。

==【完】==

看了《库存扣多了，到底怎么整》与《库存扣减多种方案》两篇文章的评论，还挺受挫的，也挺委屈的。描述了A方案和B方案，也从来没说C方案和D方案不行，好友网友反问“为什么不用C”“为什么不用D”“误导”“取关”。

只想说，时间有限，经验有限，2个小时的时间，真的只能聊1-2个技术点。

最后想起一个段子：

code review 100行代码，总能挑出100个毛病

code review 10000行代码，却只是说，代码写得不错

末了，如《库存扣多了，到底怎么整》与《库存扣减多种方案》两篇文章中很多读者的评论所述，CAS方法会遇到ABA问题，下一篇文章聊聊ABA问题，以及优化方案。

本文有收获的话，帮转下，给作者一些鼓励，谢谢。

阅读原文