链滴

MYSQL 分库分表总结

作者：bian

原文链接：https://ld246.com/article/1499225386242

来源网站：链滴

原文地址：http://wentao365.iteye.com/blog/1740874

单库单表

单库单表是最常见的数据库设计，例如，有一张用户(user)表放在数据库db中，所有的用户都可以在d库中的user表中查到。

单库多表

随着用户数量的增加，user表的数据量会越来越大，当数据量达到一定程度的时候对user表的查询会渐的变慢，从而影响整个DB的性能。如果使用mysql, 还有一个更严重的问题是，当需要添加一列的候，mysql会锁表，期间所有的读写操作只能等待。

可以通过某种方式将user进行水平的切分，产生两个表结构完全一样的user_0000,user_0001等表，user_0000 + user_0001 + ...的数据刚好是一份完整的数据。

多库多表

随着数据量增加也许单台DB的存储空间不够，随着查询量的增加单台数据库服务器已经没办法支撑这个时候可以再对数据库进行水平区分。

分库分表规则

设计表的时候需要确定此表按照什么样的规则进行分库分表。例如，当有新用户时，程序得确定将此户信息添加到哪个表中；同理，当登录的时候我们得通过用户的账号找到数据库中对应的记录，所有这些都需要按照某一规则进行。

路由

通过分库分表规则查找到对应的表和库的过程。如分库分表的规则是user_id mod 4的方式，当用户注册了一个账号，账号id的123,我们可以通过id mod 4的方式确定此账号应该保存到User_0003表中当用户123登录的时候，我们通过123 mod 4后确定记录在User_0003中。

分库分表产生的问题，及注意事项

- 分库分表维度的问题

假如用户购买了商品,需要将交易记录保存取来，如果按照用户的纬度分表，则每个用户的交易记录都存在同一表中，所以很快很方便的查找到某用户的购买情况，但是某商品被购买的情况则很有可能分在多张表中，查找起来比较麻烦。反之，按照商品维度分表，可以很方便的查找到此商品的购买情况但要查找到买人的交易记录比较麻烦。

所以常见的解决方式有：

a.通过扫表的方式解决，此方法基本不可能，效率太低了。

b.记录两份数据，一份按照用户纬度分表，一份按照商品维度分表。

c.通过搜索引擎解决，但如果实时性要求很高，又得关系到实时搜索。

- 联合查询的问题

联合查询基本不可能，因为关联的表有可能不在同一数据库中。

- 避免跨库事务

避免在一个事务中修改db0中的表的时候同时修改db1中的表，一个是操作起来更复杂，效率也会有定影响。

- 尽量把同一组数据放到同一DB服务器上

例如将卖家a的商品和交易信息都放到db0中，当db1挂了的时候，卖家a相关的东西可以正常使用。就是说避免数据库中的数据依赖另一数据库中的数据。

- 一主多备

在实际的应用中，绝大部分情况都是读远大于写。Mysql提供了读写分离的机制，所有的写操作都必对应到Master，读操作可以在Master和Slave机器上进行，Slave与Master的结构完全一样，一个Maser可以有多个Slave,甚至Slave下还可以挂Slave,通过此方式可以有效的提高DB集群的QPS.

所有的写操作都是先在Master上操作，然后同步更新到Slave上，所以从Master同步到Slave机器有定的延迟，当系统很繁忙的时候，延迟问题会更加严重，Slave机器数量的增加也会使这个问题更加重。

此外，可以看出Master是集群的瓶颈，当写操作过多，会严重影响到Master的稳定性，如果Master掉，整个集群都将不能正常工作。

所以，

1. 当读压力很大的时候，可以考虑添加Slave机器的分式解决，但是当Slave机器达到一定的数量就得虑分库了。
2. 当写压力很大的时候，就必须得进行分库操作。

MySQL使用为什么要分库分表

可以用说用到MySQL的地方,只要数据量一大, 马上就会遇到一个问题,要分库分表.

这里引用一个问题为什么要分库分表呢?MySQL处理不了大的表吗?

其实是可以处理的大表的.我所经历的项目中单表物理上文件大小在80G多,单表记录数在5亿以上,而且个表

属于一个非常核用的表:朋友关系表.

但这种方式可以说不是一个最佳方式. 因为面临文件系统如Ext3文件系统对大于大文件处理上也有许问题.

这个层面可以用xfs文件系统进行替换.但MySQL单表太大后有一个问题是不好解决: 表结构调整相关操作基

本不在可能.所以大项在使用中都会面监着分库分表的应用.

从Innodb本身来讲数据文件的Btree上只有两个锁, 叶子节点锁和子节点锁,可以想而知道,当发生页拆或是添加

新叶时都会造成表里不能写入数据.

所以分库分表还就是一个比较好的选择了.

那么分库分表多少合适呢?

经测试在单表1000万条记录一下,写入读取性能是比较好的. 这样在留点buffer,那么单表全是数据字的保持在

800万条记录以下, 有字符型的单表保持在500万以下.

如果按 100库100表来规划,如用户业务:

500万100**100 = 50000000万 = 5000亿记录.**

心里有一个数了,按业务做规划还是比较容易的.