**Mysql优化**

MySQL是一个关系型数据库管理系统，它是开源的，采用了 GPL 协议

MySQL支持大型数据库，支持 5000 万条记录的数据仓库，32 位系统表文件最大可支持 4GB，64 位系统支持最大的表文件为 8TB。

优化mysql，首先我们要对mysql逻辑架构有一定的了解。你可以把mysql服务端想象成一个大的容器，里面有四层结构，当一个请求过来后，将会执行这四层，执行一遍后才会返回给我们想要的结果。

1，首先是最上层连接层，里面是一些客户端和连接服务，主要完成一些类似于连接处理、授权认证、以及相关的安全方案，还引入了线程池的概念，为通过认证，安全接入的客户端提供线程。同样在该层上可以实现基于SSL的安全链接。服务器也会为安全接入的每个客户端验证它所具有的操作权限。

2，然后是服务层，服务层是负责解析查询(或者说编译SQL)，并且对sql进行优化(如调整表的读取顺序，选择合适的索引等)。对于查询语句，在解析查询前，服务器会先检查查询缓存，如果能在缓存中找到对应的查询结果，就不需要去进行查询解析、优化等过程，而是直接返回查询结果。其他的存储过程、触发器、视图等等都在这一层实现。

3，第三层存储引擎层，存储引擎是真正负责MySQL中数据的存储和提取，服务器通过API与存储引擎进行通信。不同的存储引擎具有的功能不同，这样我们可以根据自己的实际需要进行选取。

4，最后是数据存储层，它主要是将数据存储在运行在裸设备的文件系统之上，并完成和存储引擎的交互。**存储引擎是基于表的，而不是数据库**

在了解了mysql的一些简单的架构逻辑之后，我们要知道为什么有进行SQL优化。

SQL优化的原因有很多，比如说，**性能低、执行和等待时间太长、SQL语句写的不好、关联查询太多(设计缺陷或者一些不得已的需求)、索引失效、服务器参数设置不合理**。而且需要强调的是SQL优化的主要目标是优化索引

MySQL官方对索引的定义是：**索引是帮助MySQL高效获取数据的数据结构。所以索引的本质其实就是数据结构**。类似于字典中的目录，都是用来帮助我们快速的查找到我们想要的数据

在我们的数据表的数据之外，数据库系统还维护者满足特定查找算法的数据结构，这些数据结构以某种方式指向数据，这样就可以在这些数据结构上实现高级查找算法，这些数据结构就是索引。

由于索引本身的大小也很大，所以基本上不会全部存储在内存中，因此索引往往以索引文件的形式存储的磁盘上，我们平时所说的索引，一般都是指B树(多路搜索树，并不一定是二叉树)结构组织的索引。比如复合索引，唯一索引，聚集索引，次要索引，覆盖索引，前缀索引默认都是使用B+树索引，统称都称为索引。当然,除了B+树这种类型的索引之外，还有比如哈希索引等。

在mysql中我们可以通过SQL语句来对索引进行一些操作比如说：

创建 create index 索引名称 on 表名(创建索引的字段名);

删除 drop index 索引名称 on 表名

查看 show index from 表名

我们在创建索引的时候，也要考虑一下这个表有没有创建索引的必要：首先索引有：主键索引，唯一索引（列的值必须唯一，但允许有空值），单值索引（只包含单个列，一个表可以有多个单列索引），复合/组合索引（一个索引包含多个列）四种类型，一般情况下主键会自动建立唯一索引；频繁作为查询条件的字段应该创建索引；排序，统计，分组字段以及表关联字段和外键也可以建立索引；其中组合索引性价比更高。但是如果表记录太少或者经常进行增删改的表或者字段就不建议创建索引了，不进行条件查询的以及字段值重复的比较多的也不建议创建索引。

那么我们创建索引有什么用呢：索引的优点主要就是提高我们查询的效率，降低数据库的io交互成本，以及通过索引来进行对数据的排序，降低排序的成本，CPU的消耗等等。

索引的缺点也很明显，虽然它提高了查询的速度，但是也会降低表的更新速度，因为每次更新表的时候都要更新一下索引的字段，其实索引也是一张表它保存了主键和索引字段，并且指向实体表的记录，所以索引列也是要占用空间的，还有就是索引只是高效率的一个因素，如果数据量非常大的时候，在创建索引的同时还要对SQL语句进行优化。

在使用的过程中，有些情况下索引也会不起作用，首先索引在使用的时候要遵守最左前缀法则。最左前缀法则指的是查询从索引的最左前列开始并且不跳过索引中的列。and忽略左右关系使用索引

其次在索引的使用过程中我们用尽量避免一些会使索引失效的操作，

比如

1，在索引列上做计算，函数，类型转换，等操作，这会导致索引失效从而进行全表扫描。

2，尽量使用覆盖索引，就是说要查询的列最好是索引列，尽量不要使用select \*操作。

3，使用!=比较数值类型或者><比较字符类型的时候有可能会导致某些索引不可用。

4，is null,is not null也无法使用索引。

5，like以通配符开头如果%在索引列的左或者左右两侧会使索引失效进行全表扫描，当%在索引右边的时候则索引可以使用

6，少用or,用它来连接时会索引失效

索引使用一般性建议

尽可能通过分析信息和调整sql语句的写法来达到选择合适索引的目的，比如说在使用单键索引时，尽量选择针对当前sql语句过滤性更好的索引，在选择组合索引的时候，当前SQL语句中过滤性最好的字段，它在索引字段中的顺序，越往前越好，尽量选择可以能包含当前sql语句中的Where子句中更多字段的索引

我们可以通过Explain来查看SQL执行和索引的优化的一些信息

通过Explain+sql语句来使用

**分库分表 （会就说，不会不说）**

顾名思义，即把存于一个库的数据分散到多个库中，把存于一个表的数据分散到多个表中。分库分表 的主要原因就是随着项目的不断运行，数据库中的表以及表中的数据量都会越来越多，就会出现一些问题：我们数据库的存储资源是有限的，其负载能力也是有限的，大量的数据积累肯定会导致其处理数据的能力下降,数据量越多，那么对数据的增删改查等操作的开销也会越来越大，

分库分表的方式有

垂直分表：将一个表按照字段分成多表，每个表存储其中一部分字段。

水平分表：是在同一个数据库内，把同一个表的数据按一定规则拆到多个表中，字段不变。

垂直分库：是指按照业务将表进行分类，然后分到不同的数据库上，每个库可以放在不同的服务器上，它的核心理念是专库专用

水平分库：是把同一个表的数据按一定规则拆到不同的数据库中，每个库可以放在不同的服务器上