

优化的方式有：

1、sql 优化 ( [SQL优化的方式](#) )

2、架构层面

随着业务量越来越大，单台数据库服务器性能已无法满足业务需求，该考虑加机器了，该做集群了~~~。主要思想是分解单台数据库负载，突破磁盘I/O性能，热数据存放缓存中，降低磁盘I/O访问频率。

### 主从复制与读写分离

因为生产环境中，数据库大多都是读操作，所以部署一主多从架构，主数据库负责写操作，并做双击热备，多台从数据库做负载均衡，负责读操作，主流的负载均衡器有LVS、HAProxy、Nginx。

怎么来实现读写分离呢？大多数企业是在代码层面实现读写分离，效率比较高。另一个种方式通过代理程序实现读写分离，企业中应用较少，常见代理程序有MySQL Proxy、Amoeba。在这样数据库集群架构中，大大增加数据库高并发能力，解决单台性能瓶颈问题。如果从数据库一台从库能处理2000 QPS，那么5台就能处理1w QPS，数据库横向扩展性也很容易。

有时，面对大量写操作的应用时，单台写性能达不到业务需求。如果做双主，就会遇到数据库数据不一致现象，产生这个原因是在应用程序不同的用户会有可能操作两台数据库，同时的更新操作造成两台数据库数据发生冲突或者不一致。在单库时MySQL利用存储引擎机制表锁和行锁来保证数据完整性，怎样在多台主库时解决这个问题呢？有一套基于perl语言开发的主从复制管理工具，叫MySQL-MMM ( Master-Master replication manager for Mysql, Mysql主主复制管理器 )，这个工具最大的优点是在同一时间只提供一台数据库写操作，有效保证数据一致性。

主从复制博文：<http://lizhenliang.blog.51cto.com/7876557/1290431>

读写分离博文：<http://lizhenliang.blog.51cto.com/7876557/1305083>

MySQL-MMM博文：<http://lizhenliang.blog.51cto.com/7876557/1354576>

### 增加缓存

给数据库增加缓存系统，把热数据缓存到内存中，如果缓存中有要请求的数据就不再去数据库中返回结果，提高读性能。缓存实现有本地缓存和分布式缓存，本地缓存是将数据缓存到本地服务器内存中或者文件中。分布式缓存可以缓存海量数据，扩展性好，主流的分布式缓存系统有memcached、redis，memcached性能稳定，数据缓存在内存中，速度很快，QPS可达8w左右。如果想数据持久化就选择用redis，性能不低于memcached。

工作过程：

### 分库

分库是根据业务不同把相关的表切分到不同的数据库中，比如web、bbs、blog等库。如果业务量很大，还可将切分后的库做主从架构，进一步避免单个库压力过大。

### 分表

数据量的日剧增加，数据库中某个表有几百万条数据，导致查询和插入耗时太长，怎么能解决单表压力呢？你就该考虑是否把这个表拆分成多个小表，来减轻单个表的压力，提高处理效率，此方式称为分表。

分表技术比较麻烦，要修改程序代码里的SQL语句，还要手动去创建其他表，也可以用merge存储引擎实现分表，相对简单许多。分表后，程序是对一个总表进行操作，这个总表不存放数据，只有一些分表的关系，以及更新数据的方式，总表会根据不同的查询，将压力分到不同的小表上，因此提高并发能力和磁盘I/O性能。

分表分为垂直拆分和水平拆分：

垂直拆分：把原来的一个很多字段的表拆分多个表，解决表的宽度问题。你可以把不常用的字段单独放到一个表中，也可以把大字段独立放一个表中，或者把关联密切的字段放一个表中。

水平拆分：把原来一个表拆分成多个表，每个表的结构都一样，解决单表数据量大的问题。

## 分区

分区就是把一张表的数据根据表结构中的字段（如range、list、hash等）分成多个区块，这些区块可以在一个磁盘上，也可以在不同的磁盘上，分区后，表面上还是一张表，但数据散列在多个位置，这样一来，多块硬盘同时处理不同的请求，从而提高磁盘I/O读写性能，实现比较简单。

注：增加缓存、分库、分表和分区主要由程序猿来实现。