分表管理:

1.mysql集群:利用mysql cluster,mysql proxy,mysql replication,drdb 等等.

做mysql集群主要是减少了sql排队队列中sql的数量

优点:扩展性好,对于php代码没有分表后的复杂操作,

缺点:单个表的数量还是没有变,一次操作所花的时间还是那么多,硬件开销大.

2.预先估计会出现的大数据并且访问频繁的表,将其分为若干表

微信信息表,新建100个,根据用户的id来判断将这个用户的聊天信息放到那张表中,

数据存放的数据表 = 分表字段的内容%分表数量.

分表能够解决单表数据量过大带来的查询效率下降的问题,但不能给数据库的并发访问带来质的提升,面对高并发的写访问,我们需要对数据库进行拆分,来提高数据库的写入能力,

分库采用对关键字取模的方式,对数据库进行路由,

数据库=分库字段的内容%数据库的数量

分表分库策略:

中间变量 = 关键字%(数据库数量\*单库数据表的数量)

库 = 取整(中间变量/单库数据表的数量)

表 = (中间变量%单库数据表数量)

分表的几种方式:

1.mysql 集群事实上它并不是分表,但他可以分担数据库的操作次数,将任务分担到堕胎数据库中,集群可以读写分离,减轻读写压力,从而提升数据库的性能.

2.自定义规则分表

大表可以根据业务的规则来分解成为多个子表,

分表的规则和分区的规则一样也分为range,list,hash,key,复合模式.

===============================================================================

以聊天信息为例,

新建10张表,message00, message01, message02, message03, message04, message05, message06, message07, message08, message09

一个字段取到器hash值,取余-00

简单java代码:

String userId =UUID.randomUUID().toString();

System.out.println(userId+"hash值:"+userId.hashCode());

System.out.println("这个用户的信息将要放置的表为:"+userId.hashCode()%10);

优点:避免一张表出现几百万条数据,缩短了一条sql的执行时间,

缺点:当一种规则确定以后,打破这条规则就很麻烦,修改规则,数据存储出差,扩展性很差,

3.利用merge存储引擎来实现分表

这种情况适用于,实现没有考虑,现在已经出现了查询慢的情况.

此处创建表1,表2,合并表1和2成为表3,合并后主键有冲突,然后合并时是先创建表,再将数据插入表中,此时原来的主键会一致,在表3中插入数据,在原来的表2中会存在插入3表中的数据.

此时更新表3中的数据,会同时修改表1表2 中的数据 .

一个merge表不能再真个表上维持unique约束.当插入一个数据时,数据进入myisam表(取决于insert\_method的值),会确保在那张表中唯一,单不是跨集合的所有表.

当创建一个merge表时,没有检查底层表的存在,具有相同的表结构,当merge表被使用时,mysql会检查每个被映射的表的记录长度是够相等,不可靠.

优点:扩展性好,程序代码改动不大,

缺点:效果比第二种差