MySQL分库分表功能浅析

分库分表从字面上简单理解，就是把原本存储于一个库的数据分块存储到多个库上，把原本存储于一个表的数据分块存储到多个表上。单表操作数据量有最优值，mysql为1000万左右；可以减轻数据库的压力，不用所有线程都查同一个数据库。

数据库中的数据量不一定是可控的，在未进行分库分表的情况下，随着时间和业务的发展，库中的表会越来越多，表中的数据量也会越来越大，相应地，数据操作，增删改查的开销也会越来越大；另外，由于无法进行分布式式部署，而一台服务器的资源（CPU、磁盘、内存、IO等）是有限的，最终数据库所能承载的数据量、数据处理能力都将遭遇瓶颈。

分库分表有垂直切分和水平切分两种。

何谓垂直切分，即将表按照功能模块、关系密切程度划分出来，部署到不同的库上。例如，我们会建立定义数据库workDB、商品数据库payDB、用户数据库userDB、日志数据库logDB等，分别用于存储项目数据定义表、商品定义表、用户数据表、日志数据表等。

何谓水平切分，当一个表中的数据量过大时，我们可以把该表的数据按照某种规则，例如userID散列，进行划分，然后存储到多个结构相同的表，和不同的库上。例如，我们的userDB中的用户数据表中，每一个表的数据量都很大，就可以把userDB切分为结构相同的多个userDB：part0DB、part1DB等，再将userDB上的用户数据表userTable，切分为很多userTable：userTable0、userTable1等，然后将这些表按照一定的规则存储到多个userDB上。

应该使用哪一种方式来实施数据库分库分表，这要看数据库中数据量的瓶颈所在，并综合项目的业务类型进行考虑。

如果数据库是因为表太多而造成海量数据，并且项目的各项业务逻辑划分清晰、低耦合，那么规则简单明了、容易实施的垂直切分必是首选。

而如果数据库中的表并不多，但单表的数据量很大、或数据热度很高，这种情况之下就应该选择水平切分，水平切分比垂直切分要复杂一些，它将原本逻辑上属于一体的数据进行了物理分割，除了在分割时要对分割的粒度做好评估，考虑数据平均和负载平均，后期也将对项目人员及应用程序产生额外的数据管理负担。

在现实项目中，往往是这两种情况兼而有之，这就需要做出权衡，甚至既需要垂直切分，又需要水平切分。我们的游戏项目便综合使用了垂直与水平切分，我们首先对数据库进行垂直切分，然后，再针对一部分表，通常是用户数据表，进行水平切分。

分库分表存在的问题。

首先，在执行分库分表之后，由于数据存储到了不同的库上，数据库事务管理出现了困难。如果依赖数据库本身的分布式事务管理功能去执行事务，将付出高昂的性能代价；如果由应用程序去协助控制，形成程序逻辑上的事务，又会造成编程方面的负担。

跨库跨表的join问题。

其次，在执行了分库分表之后，难以避免会将原本逻辑关联性很强的数据划分到不同的表、不同的库上，这时，表的关联操作将受到限制，我们无法join位于不同分库的表，也无法join分表粒度不同的表，结果原本一次查询能够完成的业务，可能需要多次查询才能完成。

除此之外，额外的数据管理负担，最显而易见的就是数据的定位问题和数据的增删改查的重复执行问题，这些都可以通过应用程序解决，但必然引起额外的逻辑运算，例如，对于一个记录用户成绩的用户数据表userTable，业务要求查出成绩最好的100位，在进行分表之前，只需一个order by语句就可以搞定，但是在进行分表之后，将需要n个order by语句，分别查出每一个分表的前100名用户数据，然后再对这些数据进行合并计算，才能得出结果。