**解决高并发：**

1.减少数据库访问次数，

2. 文件和数据库分离（比如页面静态化），

3. 大数据分布式存储（比如：Mysql的主从配置读写分离），

4. 服务器的集群负载均衡，

5. 页面缓存的使用（对于页面使用局部页面缓存或数据缓存），

6. nosql内存数据库代替关系型数据库等

**分布式的案例**

项目中MySQL主从配置读写分离

在项目中为了提高数据库的查询速度，我们使用主从配置来实现MySQL的读写分离。但是虽然我们这么做能提高一部分的性能，假如用户的请求依旧很多，还是有访问的瓶颈，所以建议使用基于内存的非关系型数据库，比如Mongo或Redis.