

第四周课后练习

一个典型的大型互联网应用系统使用了哪些技术方案和手段，主要解决什么问题？请列举描述。

互联网系统发展的不同阶段，需要不同的投入和不同的架构，以满足不同级别的需求。一个大型互联网应用会有比较多的方案和手段以满足高并发，高可用，以及研发的可扩展性的等等需求。其核心问题还是在于高并发高性能。

下面根据不同的问题来总结有哪些解决方案：

可扩展性：

可扩展性在项目内部主要是对各种设计模式的应用，以使复杂系统项目可以满足不断变更的需求的需要。在架构层面还有以下的一些设计：

1. 分层：把服务分成多个层级，前段，网络，服务端，存储，大数据等，不同层级的服务部署在不同的集群中，并以此组成整个系统。
2. 分割：服务在进行横向的分层后，还可以进行纵向的分割，进一步把一些复杂的层切分成一个个的业务单元，这些业务单元可以由单独的部门负责，独立运营。
3. 微服务和中台化：当一个网络系统发展壮大以后，很多业务单元功能可能会在不同业务部门之间重复，比如商品，订单，发红包等等。这些公共的服务可以单独出来独立运营开发，以减少开发代价。

可伸缩性：

可伸缩性主要指的是服务水平扩展的能力，即当系统压力上升的是否服务是否能迅速通过加机器的方式使得服务能力提升。

在业务应用层面，应该让应用保持无状态，每台机器处理的任务都一样，这样可以迅速的对机器进行扩容。

在数据层面，对一些只读的数据，可以采用NoSql替代mysql，从而实现较为容易的扩容。

高并发以及性能相关：

大型网站的可用性性能是非常重要的，为了保证高并发流量下网站可以正常使用，需要通过架构设计来保证并发性能：

1. 集群：集群是最常用的增加性能的方式，当系统资源不够用时，不是垂直扩容，而是水平增加机器。
2. 缓存：通过CDN，反向代理，本地缓存，远程缓存等层层缓存的部署，大幅度减少对核心业务集群以及mysql集群的访问，以增加系统对于大流量的应对能力。

3. 主从分离，分库分表，冷热分离：针对数据库着一层，为了不让数据库性能成为高并发的瓶颈，可以用主从分离的方式将读写分离，以增加数据库的处理能力，用分库分表的方式把数据处理分散到多个服务器，用冷热分离的方式把不常用的，只读的数据丢到 Nosql 去，使得主库可以集中处理一些更为重要的功能，并增加可伸缩性。
4. 异步：系统服务之间没有直接的耦合，而是通过互相发送消息的机制共享数据，从而完成整个系统。这种方式可以起到削峰的作用，把超高并发的请求存起来，再用一个比较平缓稳定的速度去吃力。

高可用相关：

一个大型的网站会采用几十上百万台服务器来提供服务，服务器宕机就成了经常会发生的事情，因此需要在架构上做一些设计，来防止服务器宕机引起服务的不可用。

1. 冗余：通过冗余部署的方式来应对服务器宕机对系统的冲击。比如数据层需要有冗余备份，应用服务可以起多个实例，当一台服务器宕机时，可以切到其他的服务器上去。
2. 自动化：主要集中在运维方面，通过自动化部署，排除故障等，使得系统在无人值守的时候依旧可以正常运行。

安全：

通过登录认证，加密，验证码等方式防止 XSS 攻击，SQL 注入等，通过风控过滤垃圾信息，敏感信息等。