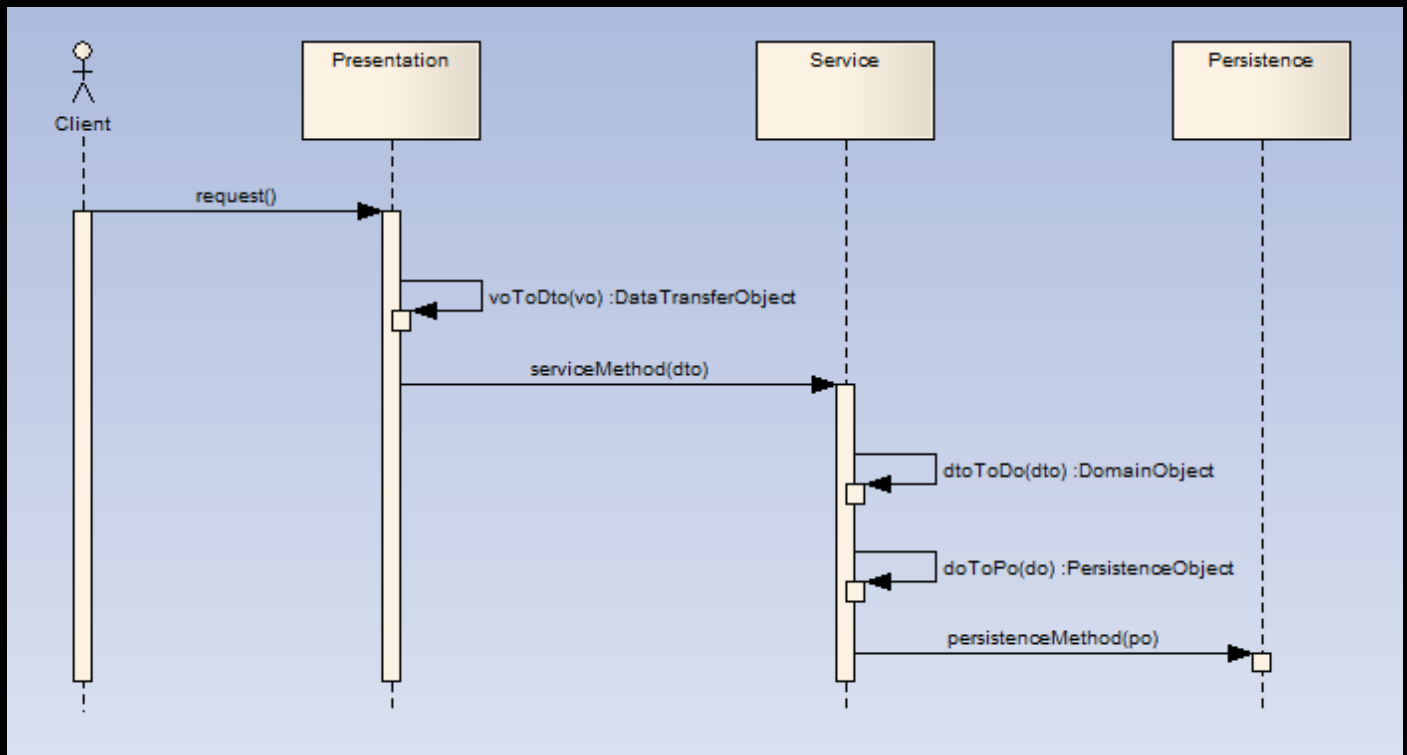


Architecture

分布式架构路线

1. **VO (View Object):** 视图对象，用于展示层，它的作用是把某个指定页面（或组件）的所有数据封装起来。
2. **DTO (Data Transfer Object):** 数据传输对象，这个概念来源于 J2EE 的设计模式，原来的目的是为了 EJB 的分布式应用提供粗粒度的数据实体，以减少分布式调用的次数，从而提高分布式调用的性能和降低网络负载，但在这里，我泛指用于展示层与服务层之间的数据传输对象。
3. **DO (Domain Object):** 领域对象，就是从现实世界中抽象出来的有形或无形的业务实体。
4. **PO (Persistent Object):** 持久化对象，它跟持久层（通常是关系型数据库）的数据结构形成一一对应的映射关系，如果持久层是关系型数据库，那么，数据表中的每个字段（或若干个）就对应 PO 的一个（或若干个）属性。



<http://www.blogjava.net/johnnylzb/archive/2010/05/27/321968.html>

集群与分布式

完全是两个东西

1. **分布式：纵向拆分**，一个业务分拆多个子业务，部署在不同的服务器上。主要是业务层面拆分，进行业务解耦，从而提高服务高可用以及高性能。
2. **集群：横向复制**，同一个业务，部署在多个服务器上，前面通过负载均衡，起到分担压力的作用。而且这些服务器中，即使有一两个宕机也不会影响到整体业务。

单机结构:



(累如狗)

全栈老实人



集群结构:



+



(凑合过)

全栈老实人

全栈老实人



分布式:



+



(男女搭配干活不累)

后端攻城狮

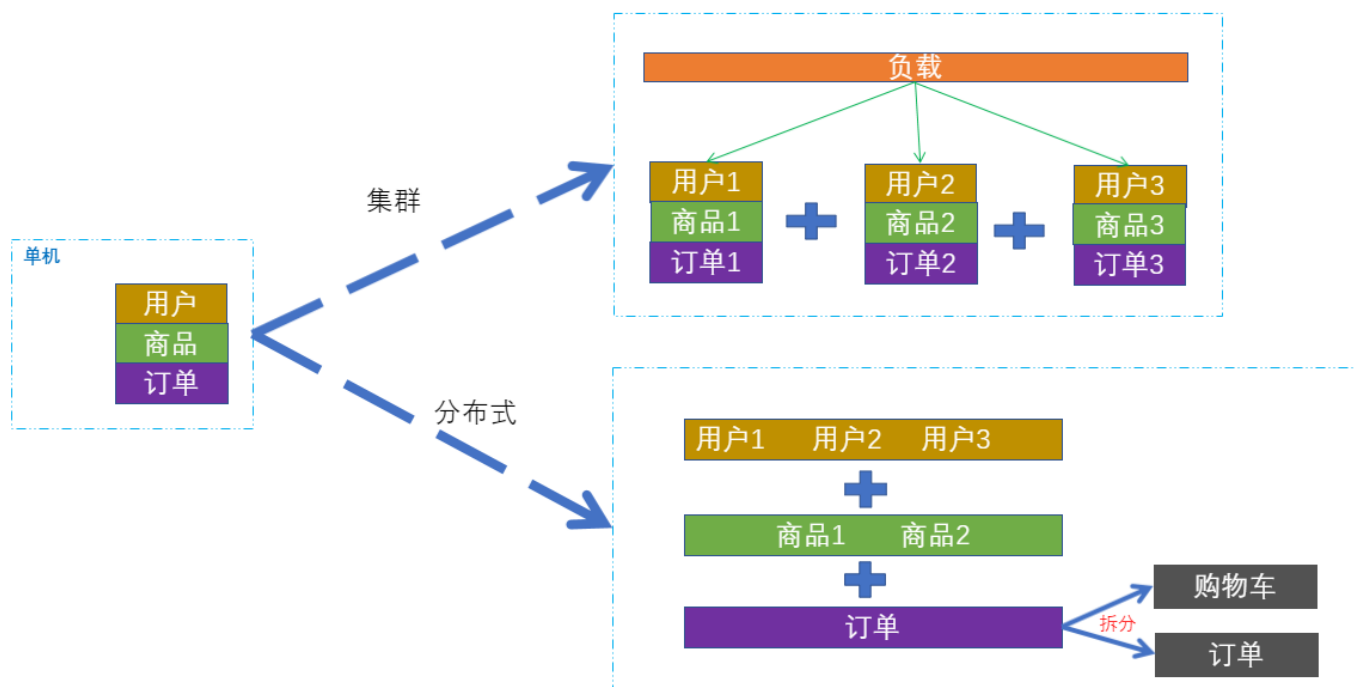
前端攻城狮

横向复制

纵向切分

[log.csdn.net/HeatDeath](https://blog.csdn.net/HeatDeath)

<https://blog.csdn.net/b379685397>



<https://blog.csdn.net/b379685397>