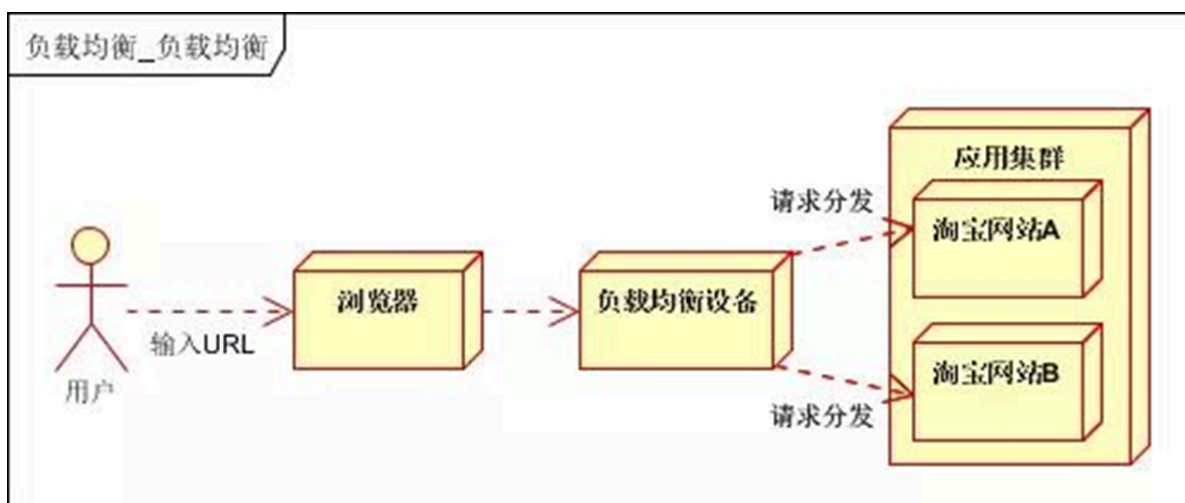


第九章 基本云架构

9.1 负载分布架构

1. 负载分布架构

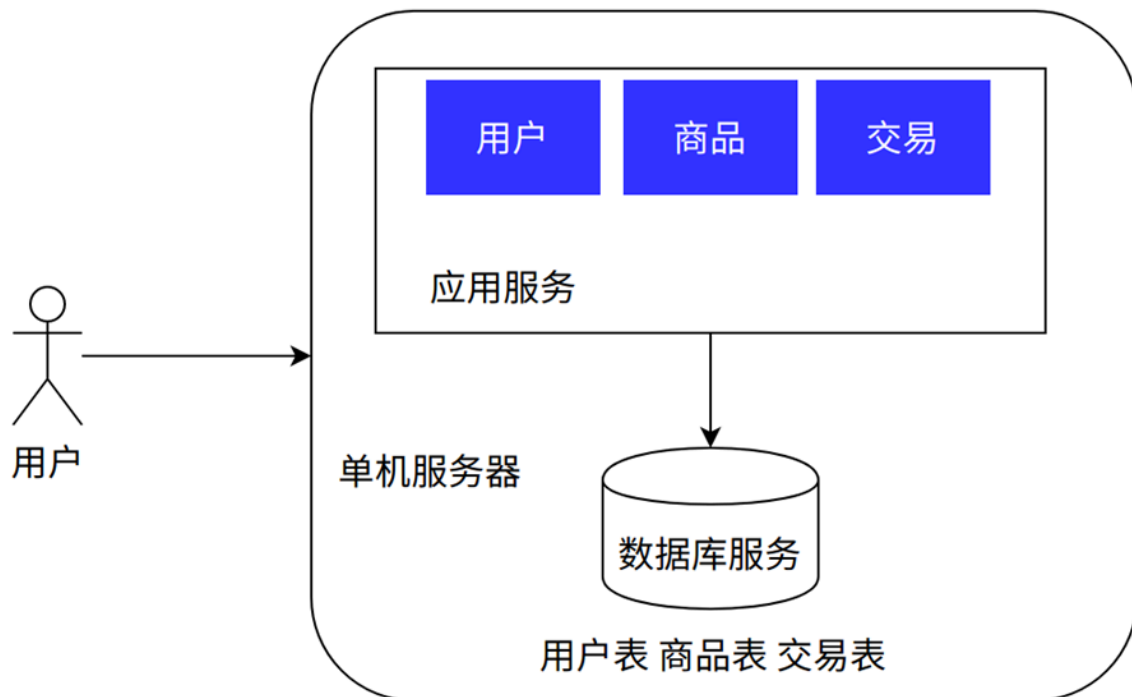
1. 通过增加一个或多个相同的IT资源可以进行IT资源**水平扩展**，而提供运行中逻辑的负载均衡器能够在可用的IT资源上均匀分配工作负载。
2. 在一定程度上依靠复杂的**负载均衡算法**和**运行时逻辑**，减少IT资源的过度使用和使用率不足的情况。
3. 负载分布常常可以用来支持**分布式虚拟服务器、云存储设备、云服务、微服务**。



4.

2. 单机系统

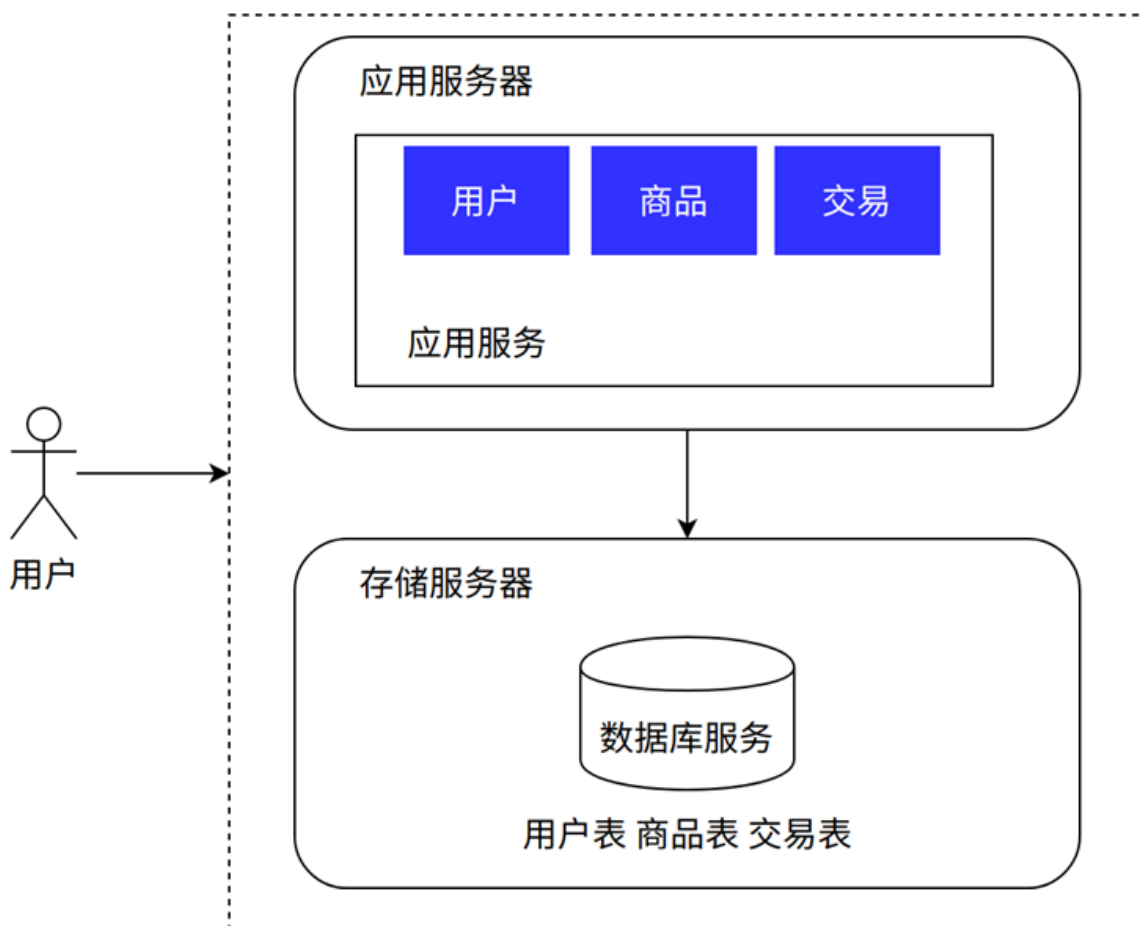
1. 所有的业务全部写在一个**项目**中，部署服务到**一台服务器**上。
2. 优点：
 1. 架构简单，易于开发和维护
 2. 没有网络通信的延迟
3. 缺点：
 1. **扩展性差**，当访问量和数据量增加时，单机资源（CPU、内存、存储）会迅速耗尽
 2. **单点故障**（SPOF）：如果服务器宕机，整个系统会不可用。



4.

3. 应用、数据库分离

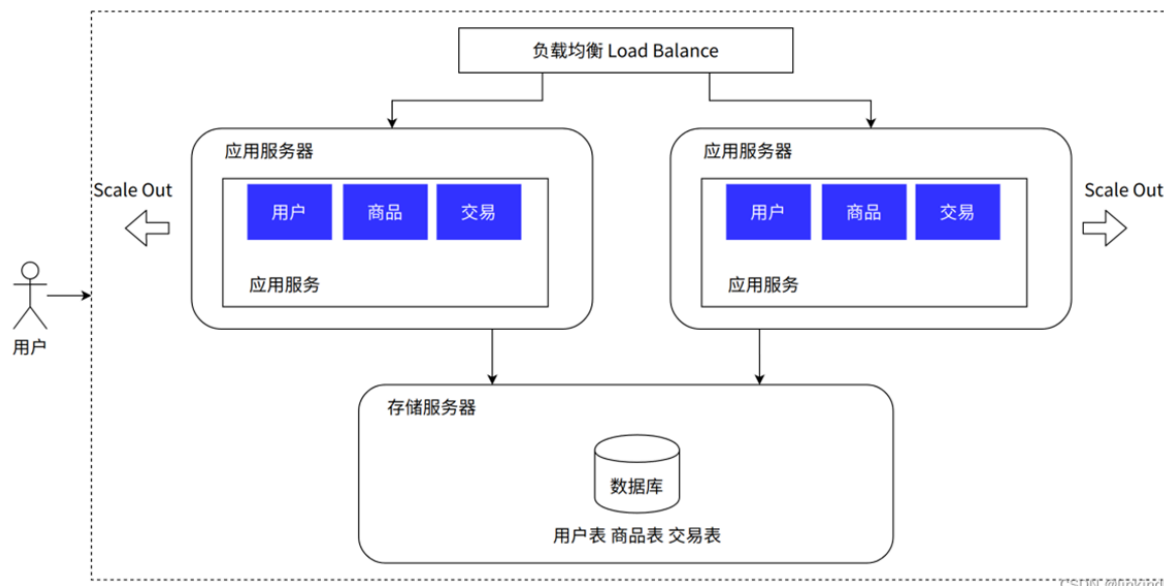
1. 为了提升性能和分担负载，Web应用开始采用**两层架构**，将Web服务器和数据库服务器分离。
2. 优点：提供性能和拓展性；增强了模块化和维护性。
3. 缺点：单点故障、性能存在瓶颈。



4.

4. 应用服务集群

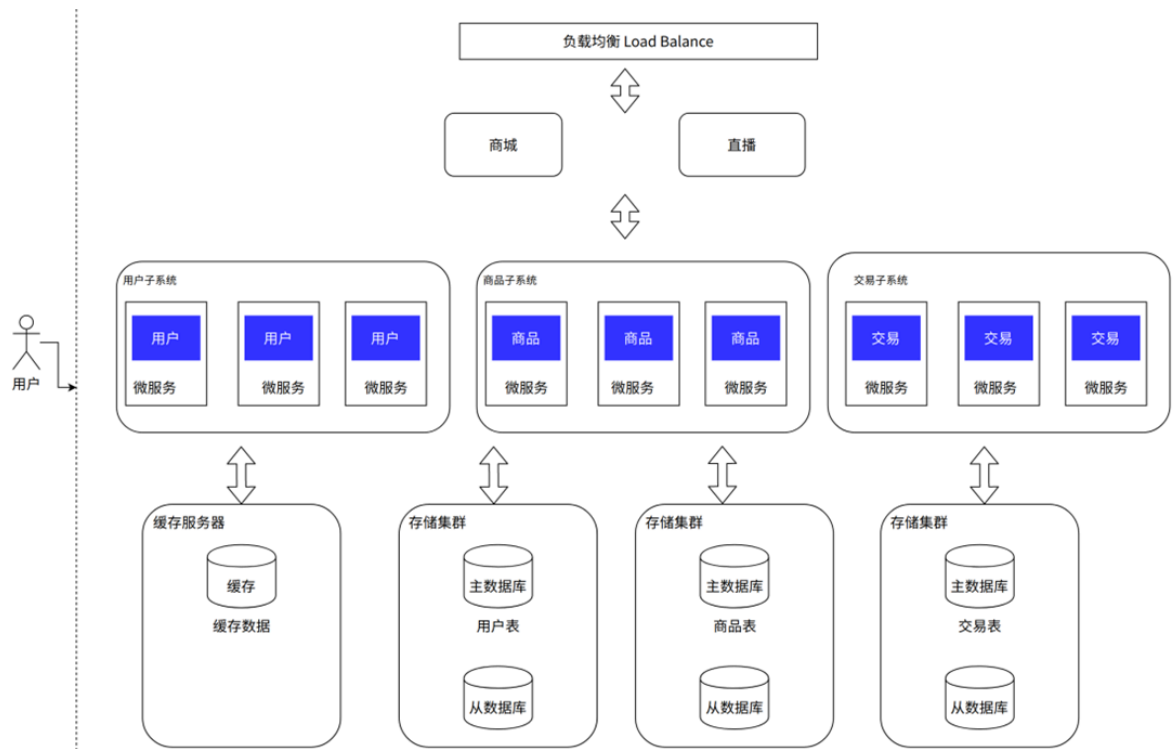
1. 增加**应用层服务器**，将用户流量分担到不同的应用层服务器上，来提升系统的承载能力。
2. 优点：高可用、高拓展性等。
3. 缺点：存储服务器单点故障、性能存在瓶颈。



4.

5. 微服务

1. 将应用拆分为多个独立的小服务，每个服务负责特定的功能。
2. 独立开发、部署和扩展，通过轻量级的通信协议（如HTTP/REST）进行交互。
3. 优点：
 1. 提高了系统的**灵活性和可维护性**，每个服务可以独立演化。
 2. **技术多样性**，可使用不同语言、框架、数据库进行开发
 3. 提供了更好的**故障隔离能力**，某个服务的故障不会影响整个系统。
4. 缺点：
 1. 增加了系统的**复杂性**，需要处理服务间的通信、数据一致性和分布式事务问题。
 2. 运维**成本增加**，需要完善的监控和管理工具。

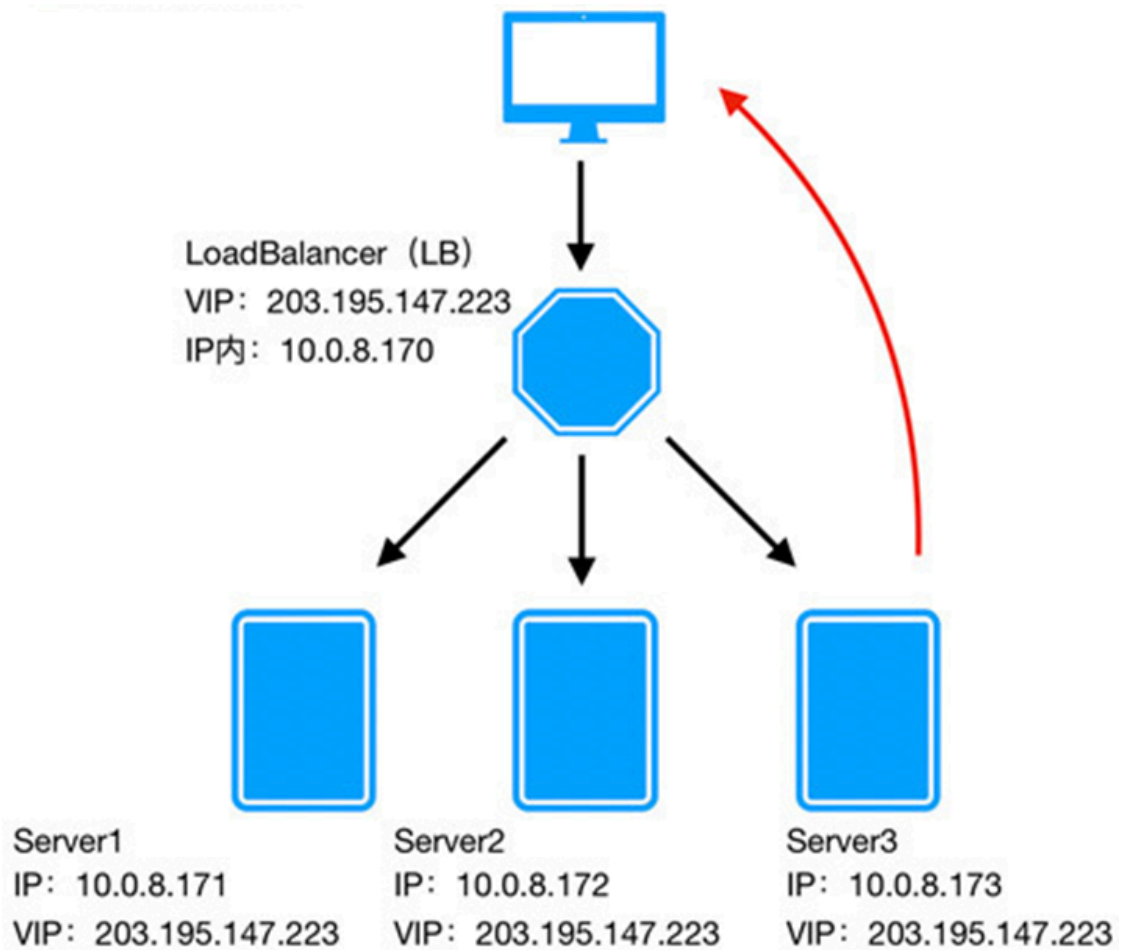


5.

6. 负载均衡可以根据网络协议的层数进行分类

1. 二层负载均衡

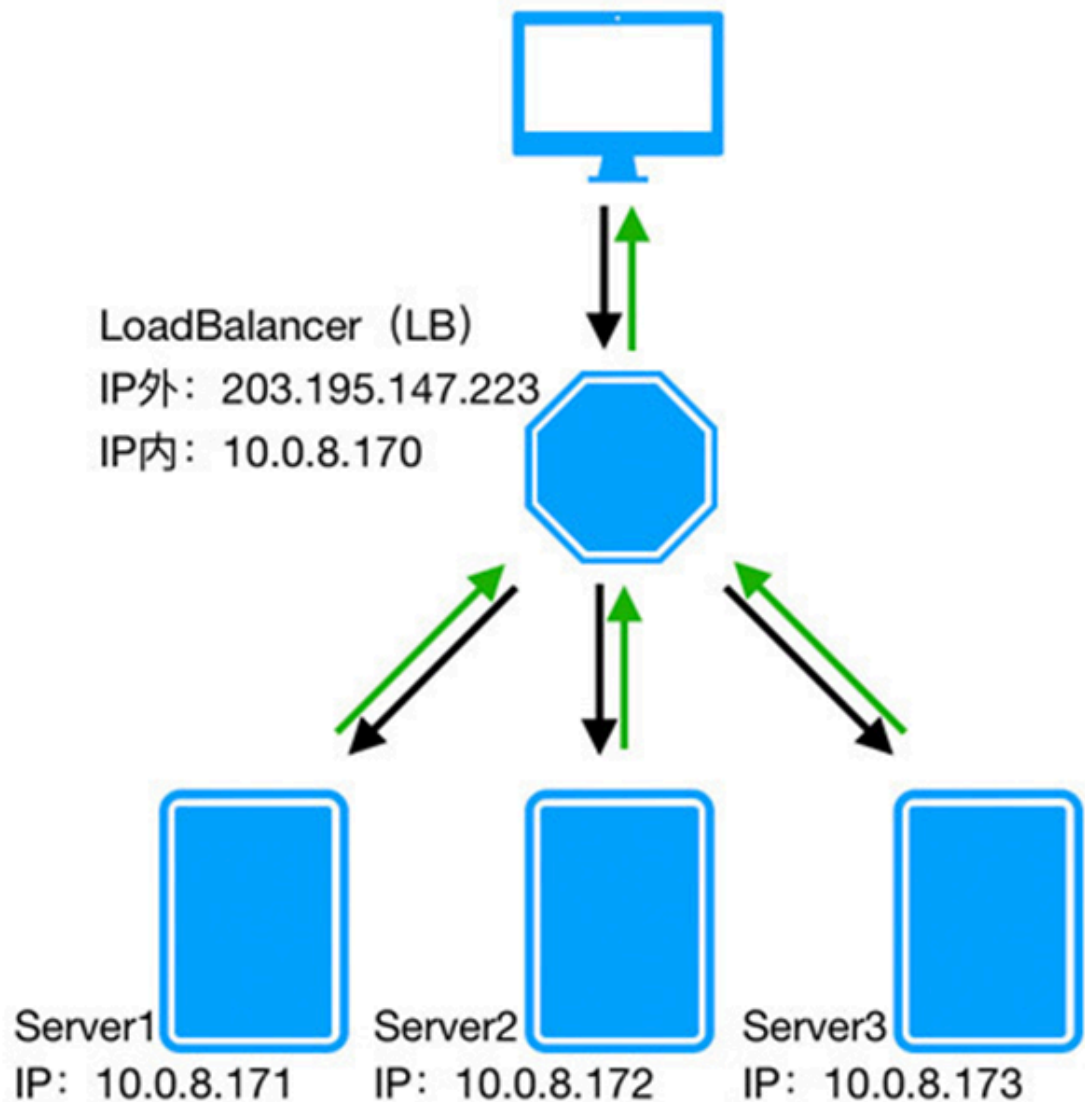
1. 负载均衡服务器对外提供一个VIP（虚拟IP），集群中不同的机器采用**相同IP地址**，但**机器的MAC地址**不一样。当负载均衡服务器接受到请求之后，通过改写报文的目标MAC地址的方式**将请求转发到目标机器**实现负载均衡。
2. 特点：控制粒度比较粗；负载均衡服务器压力小；吞吐量高



3.

2. 三层负载均衡

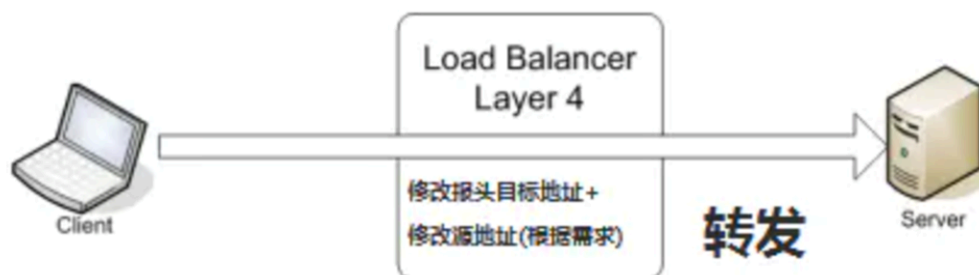
1. 负载均衡服务器对外依然提供一个VIP，但集群中不同的机器采用**不同的IP地址**。当负载均衡服务器接受到请求之后，根据不同的负载均衡算法，通过IP将请求转发至不同的真实服务器。
2. 特点： 控制粒度比较粗；请求响应经过负载均衡器服务器，其压力较大



3.

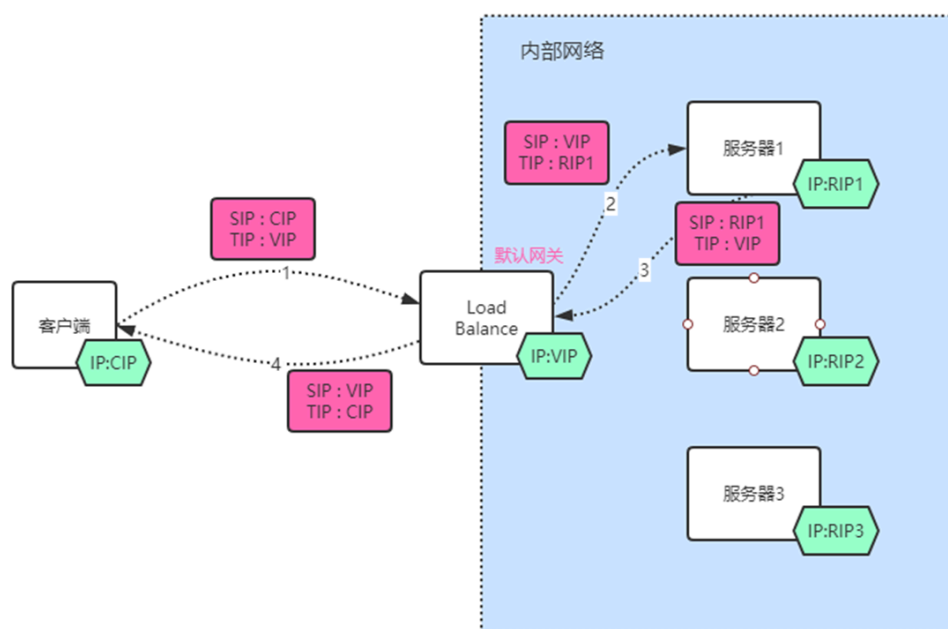
3. 四层负载均衡

1. 工作在OSI模型的传输层，该层协议除了包含**源IP、目标IP**以外，还包含**源端口号及目的端口号**。四层负载均衡服务器在接受到客户端请求后，通过修改数据包的地址信息（**IP+端口号**）将流量转发到应用服务器。
2. 根据请求处理的模式，负载均衡方式可以分为：NAT, D-NAT, DR（Direct Routing, 直接路由模式）以及TUN隧道技术等。
3. 特点： 不理解应用层协议

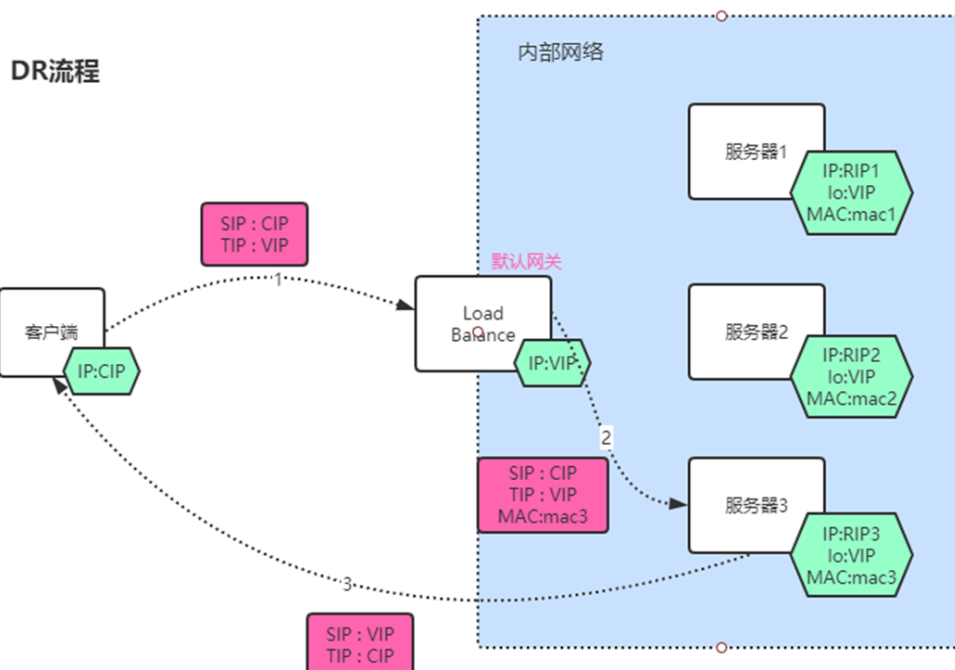


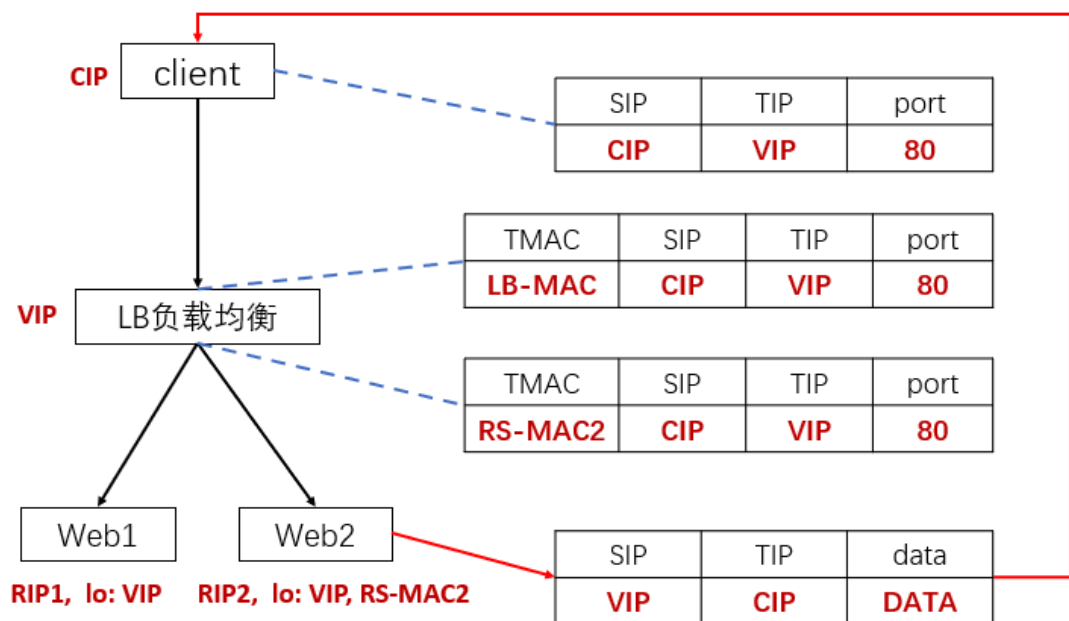
4.

5. NAT, 特点：请求响应经过负载均衡器服务器，其压力较大



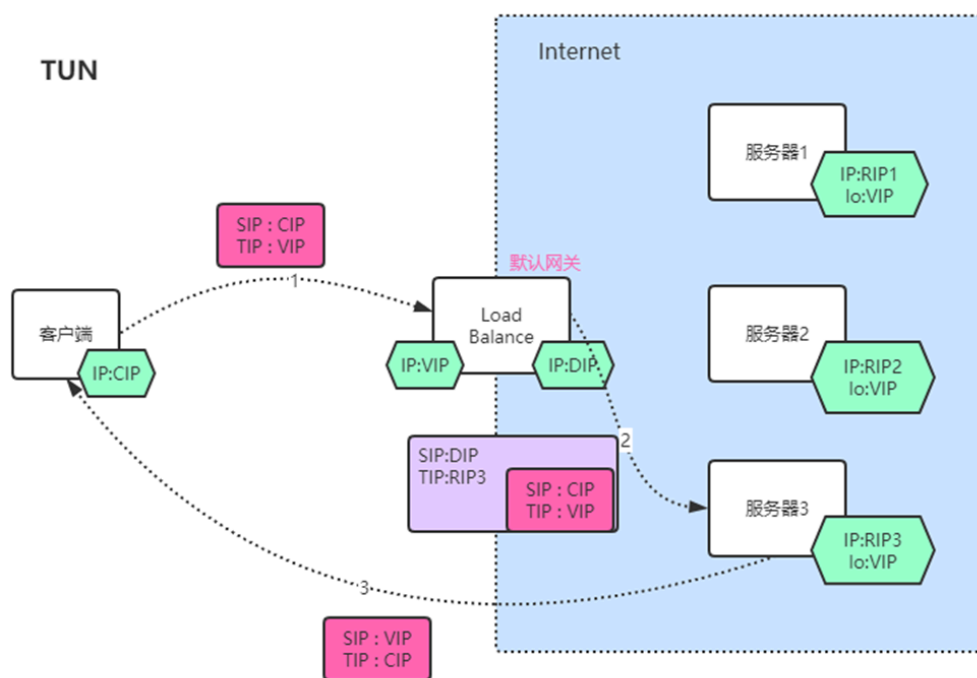
6. DR ((Direct Routing), 特点：响应不经过负载均衡器服务器，其压力较小





21

7. TUN隧道技术，特点：响应不经过负载均衡器服务器，其压力较小



4. 七层负载均衡

1. 基于应用层的负载均衡，可以根据请求的URL进行转发负载，在四层负载的基础上，考虑应用层的特征。
2. 特点：更灵活，控制粒度更细。

3. Nginx

1. Nginx是一款轻量级的高性能web服务器，同时也是一款非常优秀的负载均衡器和反向代理服务器。由于它的内存占用少，启动极快，高并发能力强，在互联网项目中广泛应用。
2. 正向代理（Forward Proxy）；反向代理（Reverse Proxy）



3.

7. 负载均衡算法

1. 负载均衡服务器在决定**将请求转发到具体哪台真实服务器**时，是通过负载均衡算法来实现的。
2. 负载均衡算法可以分为两类：静态和动态负载均衡算法。
3. 静态负载均衡算法：轮询法、随机法、源地址哈希法、加权轮询法、键值范围法。
4. 动态负载均衡算法：最少连接数、最快响应速度、观察方法、预测法等。

8. 静态负载均衡算法

1. **轮询法**：按请求顺序轮流分配。
 1. 优点：简单高效、易拓展
 2. 缺点：性能受限于木桶最短木板理论。
2. **随机法**：将请求随机分配到各个节点。
 1. 优缺点和轮询类似。
3. **源地址哈希法**：根据客户端的**IP地址**，通过哈希函数计算得到一个数值，用该数值对服务器节点数进行取模，得到的结果便是要访问节点序号。
 1. 优点：可人为干预某客户端请求的节点。
 2. 缺点：无法保证高可用性，可能导致负载不均衡。
4. **加权轮询法**：根据机器的配置分配不同的权重，将请求顺序且按照权重分配到后端。
 1. 优点：可以将不同机器的性能问题纳入到考量范围，集群性能最优最大化
 2. 缺点：无法动态调整节点权重。
5. **加权随机法**：根据后端机器的配置，系统的负载分配不同的权重。
6. **键值范围法**：根据键的范围进行负债，比如0到10万的用户请求走第一个节点服务器，10万到20万的用户请求走第二个节点服务器...
 1. 优点：容易水平拓展
 2. 缺点：容易负载不均、无法满足高可用性。

9. 动态负载均衡算法

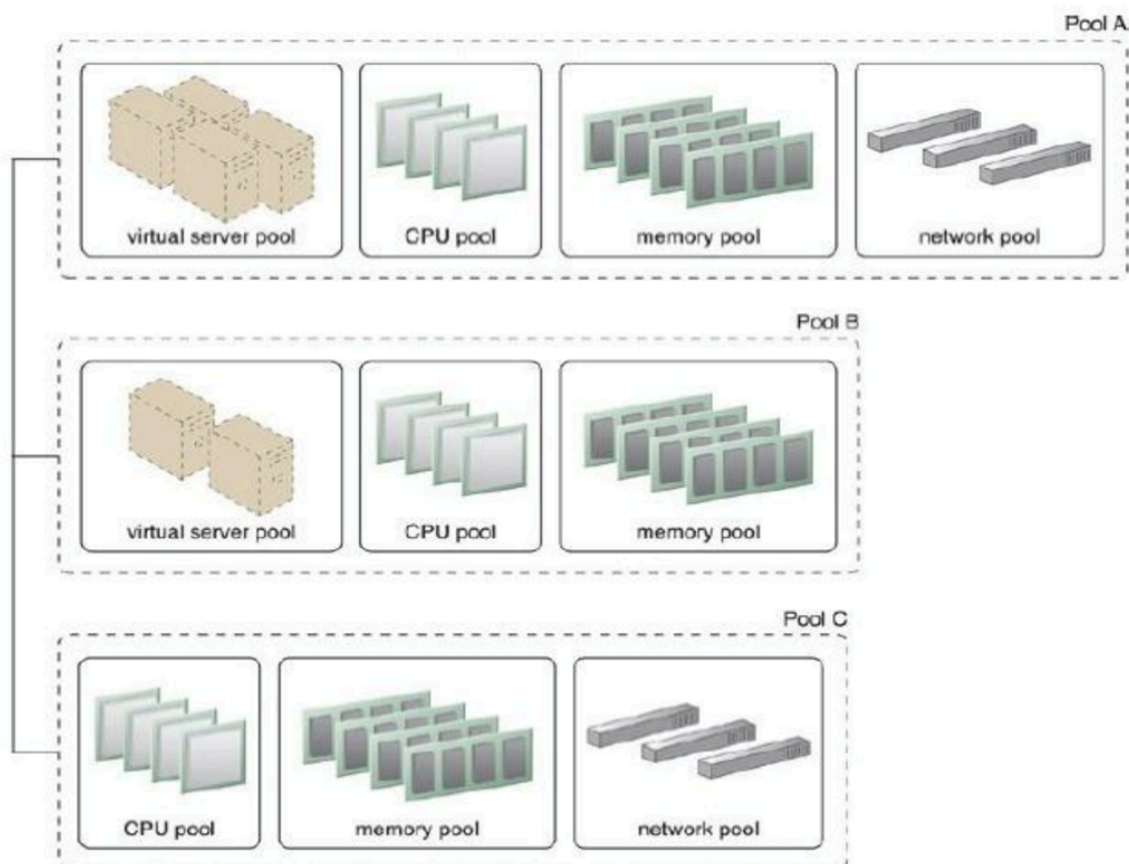
1. **最小连接数**：每一台服务器的当前连接数进行统计(L4 层连接数)，当有新的请求进入时，将新的请求分配给**当前最少连接处理的服务器**。
 1. 最小连接数为**最常用的负载均衡算法之一**，在后台服务器处理能力均等的情况下，使用最小连接数可以得到最为平衡的负载均衡效果。
2. **最快响应速度**：通过观察每台服务器的应用响应速度，当有新的请求进入的时候，将新的请求分配给响应最快的服务器。
3. **动态观察**：连接数目和响应时间以这两项的最佳平衡为依据为新的请求选择服务器。

10. 以下机制也是该云架构的一部分

1. 审计监控器：分配运行时工作负载时，就满足法律和监管要求而言，处理数据的IT资源类型和地理位置可以决定监控是否是必要的。
2. 云使用监控器：各种监控器都能参与执行运行时工作负载的跟踪与数据处理。
3. 虚拟机监控器：虚拟机监控器与其托管的虚拟服务器之间的工作负载可能需要分配。
4. 逻辑网络边界：逻辑网络边界用于隔离、确定如何分布、以及在哪里分布工作负载相关的云用户网络边界。
5. 资源集群：支持不同集群节点间的负载均衡。
6. 资源复制：生成虚拟化IT资源的新实例。

9.2 资源池架构

1. 使用一个或多个资源池为基础，相同的IT资源由一个系统进行分组和维护，确保他们保持同步。
 1. 常见的资源池有：物理服务器池、虚拟服务器池、存储池、网络池、CPU池、内存池。
2. 资源池可以建立**层次结构**，形成资源池父子、兄弟和嵌套关系，从而满足不同的资源池需求。
 1. 如，同级的相互隔离，云用户访问各自的资源池。
 2. 嵌套的可以满足组织的不同部门或不同组。



3.

9.3 动态可扩展架构

1. 基于一个预先定义的可扩展条件的系统，触发条件会导致从资源池动态分配IT资源。
2. 动态扩展类型有

1. 动态水平扩展：指分配和释放IT资源，其中分配资源称为向外扩展(scaling out)，释放资源称为向内扩展(scaling in)。
2. 动态垂直扩展：垂直扩展(vertical scaling)：指现有IT资源被具有更大或更小容量的资源代替，分别称为向上(scaling up)扩展和向下(scaling down)扩展。
3. 动态重定位：将IT资源重放置到更大容量的主机上。

9.4 弹性资源容量架构

主要与虚拟服务器的动态供给相关，该架构可以分配和回收CPU与RAM资源，从而可以立即响应服务对IT资源的需求变化。

9.5 服务负载均衡架构

1. 是工作负载分布架构的变种，是专门针对云服务实现的。在动态工作负载上增加负载均衡系统，就创建了云服务的冗余部署。
2. 云服务实现的副本被组织为一个资源池，而负载均衡器则作为外部或内置组件，允许托管服务器自行平衡工作负载。

9.6 云爆发架构

1. 只要达到预先设定的容量阈值，就从企业内部的IT资源扩展或“爆发”到云中。
2. 相应的云IT资源是冗余性预部署，它们会保持非活跃状态，直到发生云爆发。
3. 当不再需要这些资源后，基于云的IT资源被释放，架构则“爆发入”企业内部，回到企业内部环境。

9.7 弹性磁盘供给架构

一个动态存储供给系统，确保按照云用户实际使用的存储量进行精确计费。

9.8 冗余存储架构

引入了复制的辅云存储设备做为故障系统的一部分，它要与主运存储设备中的数据保持同步。当主设备失效时，就把请求转向辅设备。