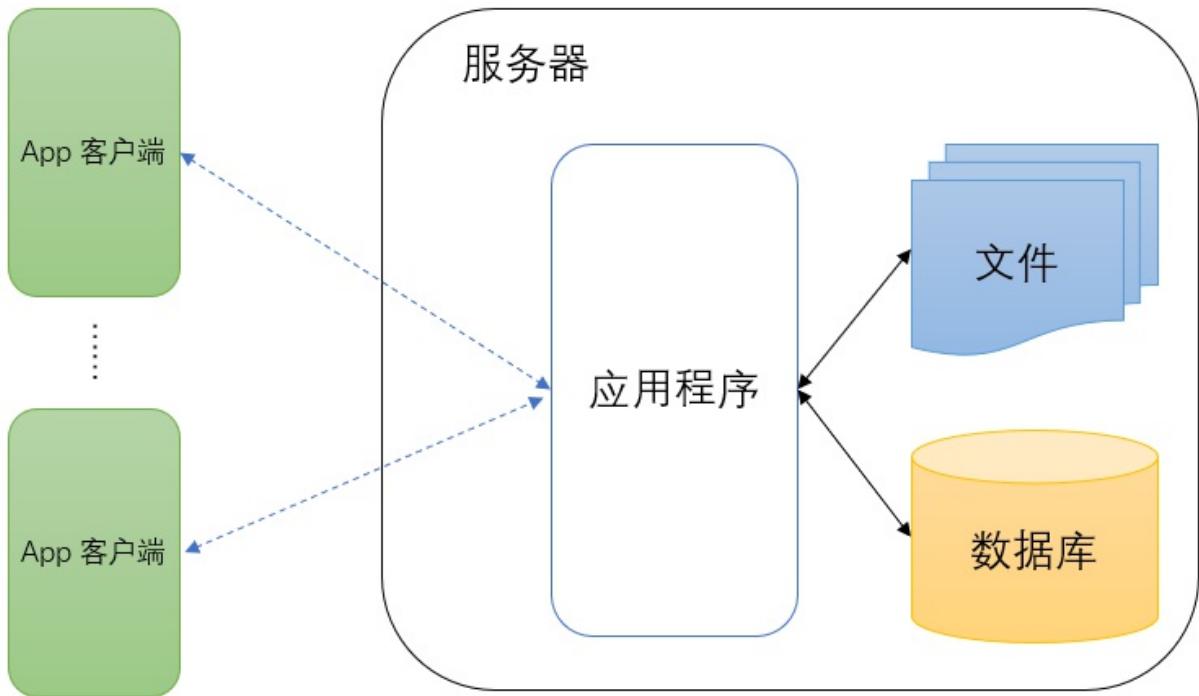


大型 HTTP 服务器架构演进路线及思路

一个成熟的大型后端服务器（如京东、淘宝等）并不是一开始的设计就具备完整的高性能、高可用、高安全等特性。它是随着业务和用户量的增长，业务功能不断地扩展演化而来的。在这个过程中，团队的增加带来开发模式的转变，性能瓶颈带来技术架构及设计思想的改变。随着业务的增长，开始出现业务功能的侧重点，如微信在发展成十亿级别的用户体量后，业务侧重的就是如何解决数十亿用户实时消息传输的通达性，百度慢慢地发展为如何处理海量数据的搜索请求。这些技术架构方案及思想，各有各的不同，无法一蹴而就，而是业务驱动。本节无法详尽提及每一种架构的设计方案，而只提供一种通用的思想，这些思路广泛应用于现在大型的后端服务器设计架构中，希望读者在具体的项目中，随着业务的增长，能从这里找到一些思路。下面将从服务器的演进路线进行讲解。

1. 项目初期

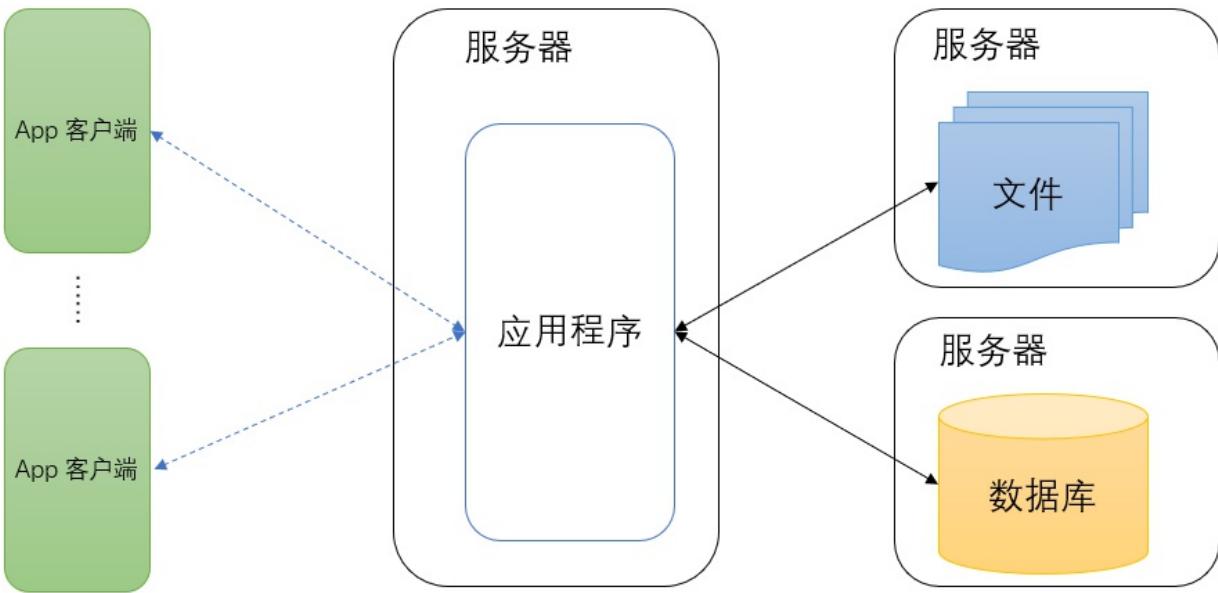
在项目的初期，一般为了抢夺市场时间窗口，产品需要快速推向市场。此时，访问量低，业务单一，对服务器要求不高。正如本小册的方式，应用程序、数据库和文件全部放在单一的服务器中，如下图所示。



但需要注意的是，单一不代表粗糙，应用程序设计阶段，应秉承解耦的思想，各业务组件之间相对独立，各层级清晰，如本小册的 views 模块分层逻辑、数据库处理模块化等。

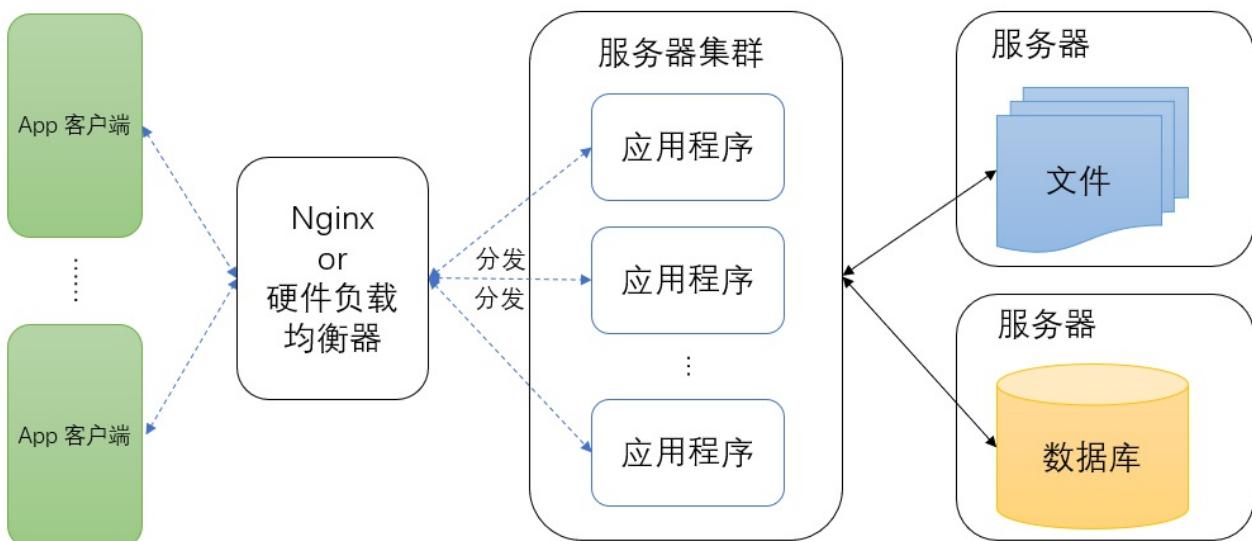
2. 应用程序、数据、存储分离

随着业务的发展，单台服务器已无法满足业务需求，此时应将应用程序、数据库和存储进行拆分，将其各自部署在不同的服务器上。



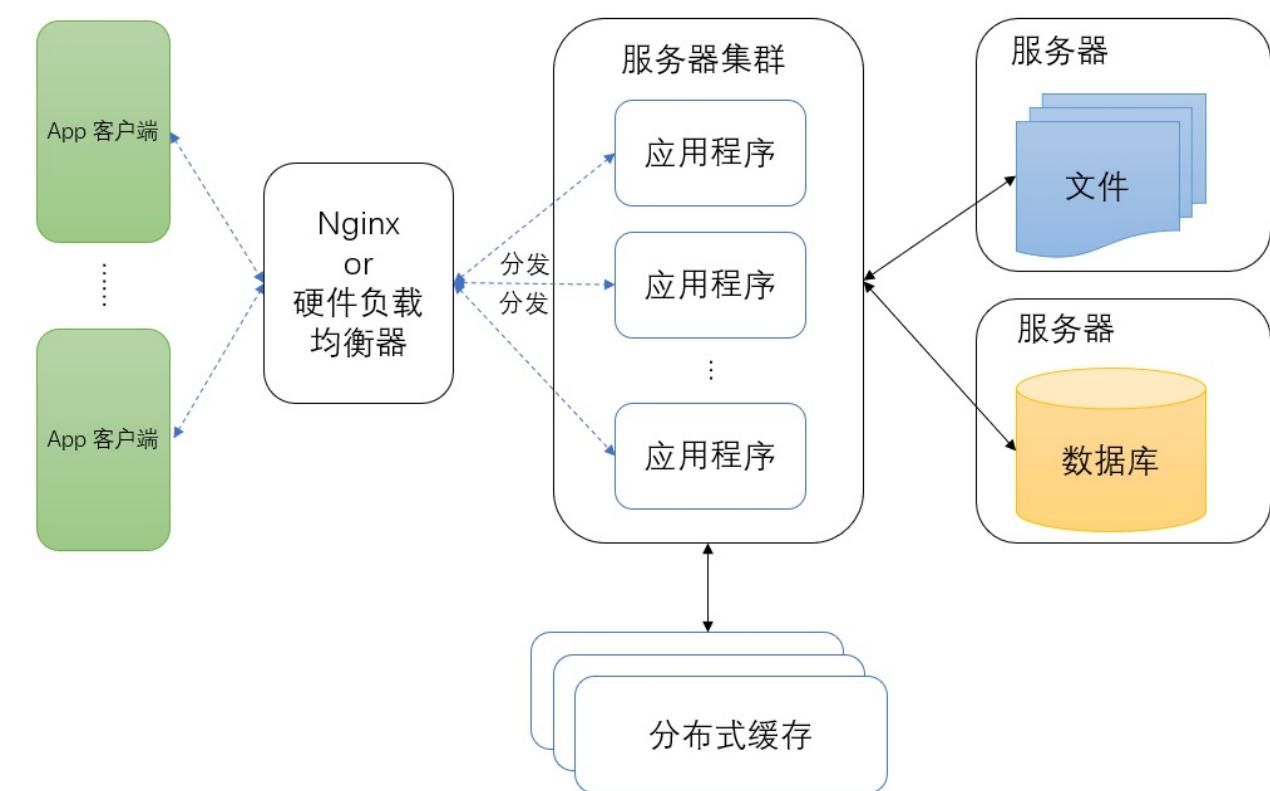
3. 负载均衡

当服务器的访问量大于单台服务器能提供的能力时，此时需要部署多台服务器进行横向扩展。在服务器集群前增加负载均衡器，以使访问流量通过负载均衡器能均衡地分配到后端服务器集群上，以此来满足大流量、高并发、海量数据请求问题。目前主流的负载均衡分软件和硬件两种，软件有主流的 Nginx，硬件需要购买专门的负载均衡器设备，成本较高，但处理能力更强。

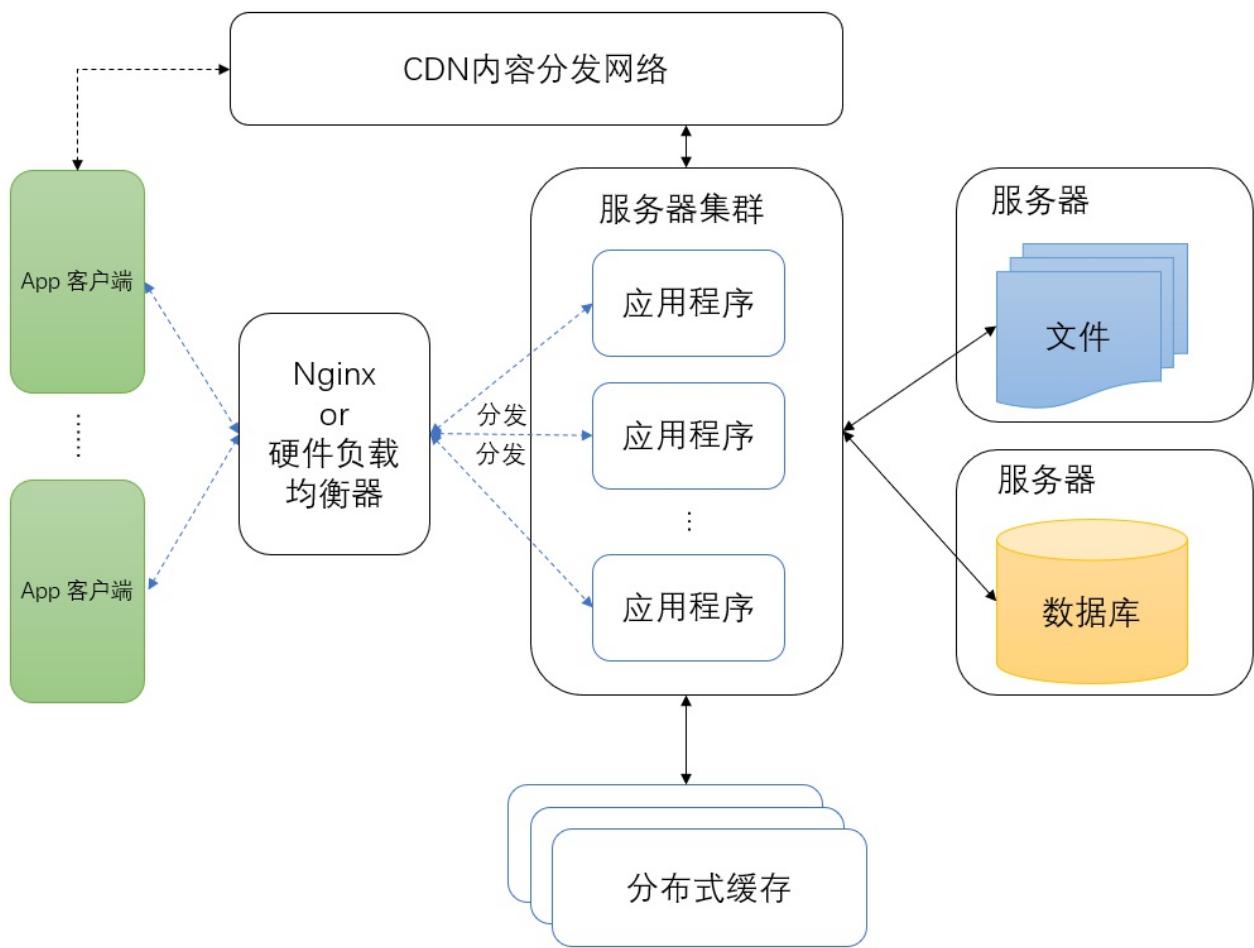


4. 缓存技术

缓存技术能大大提高服务器性能，世间万物大多遵循 2/8 原则，用在这里，即 80% 的访问量落在 20% 的业务数据上。对热点数据（20%）进行跟踪并进行缓存，能大大提高访问效率。缓存分为文件缓存、内存缓存及数据库缓存。缓存主要分两种，一是使用本地缓存，另一种是分布式缓存。本地缓存一般用于单机模式，缓存数据量有限；而分布式缓存可以缓存海量数据，易扩展，容灾性强，常用的分布式缓存有 Memcache 及 Redis。

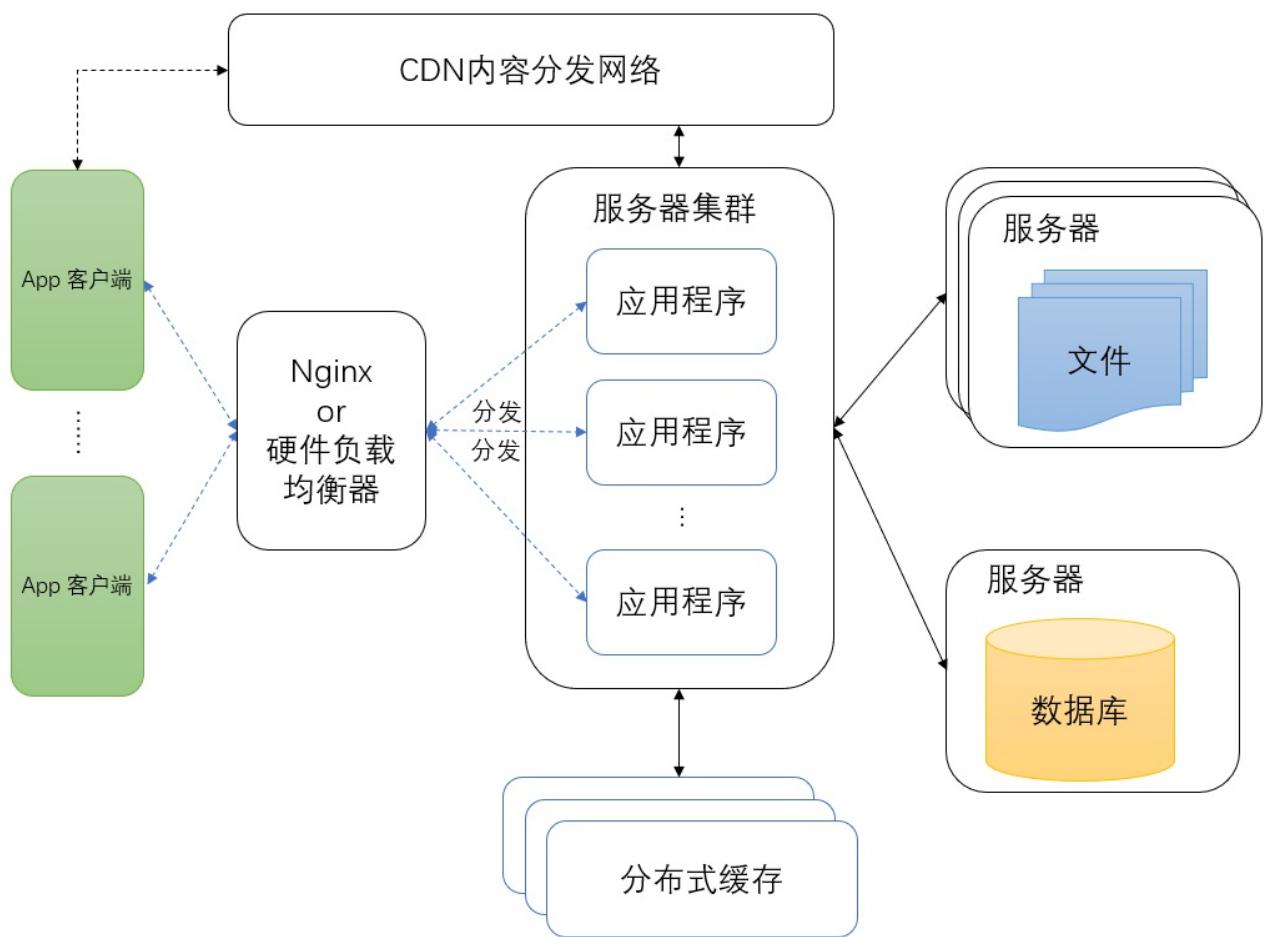


在缓存技术中，还有一个重量级的服务，叫 CDN。简单讲，即就近接入，提高用户访问速度。当目标用户分散在全国各地，此时部分用户受地域、网络等限制，访问服务器存在延迟问题，特别是点播、直播等场景。CDN 就是这样的一种技术，它能将源站点内容抓取分发到最接近用户的节点，从而提高用户的访问速度和提升用户体验。



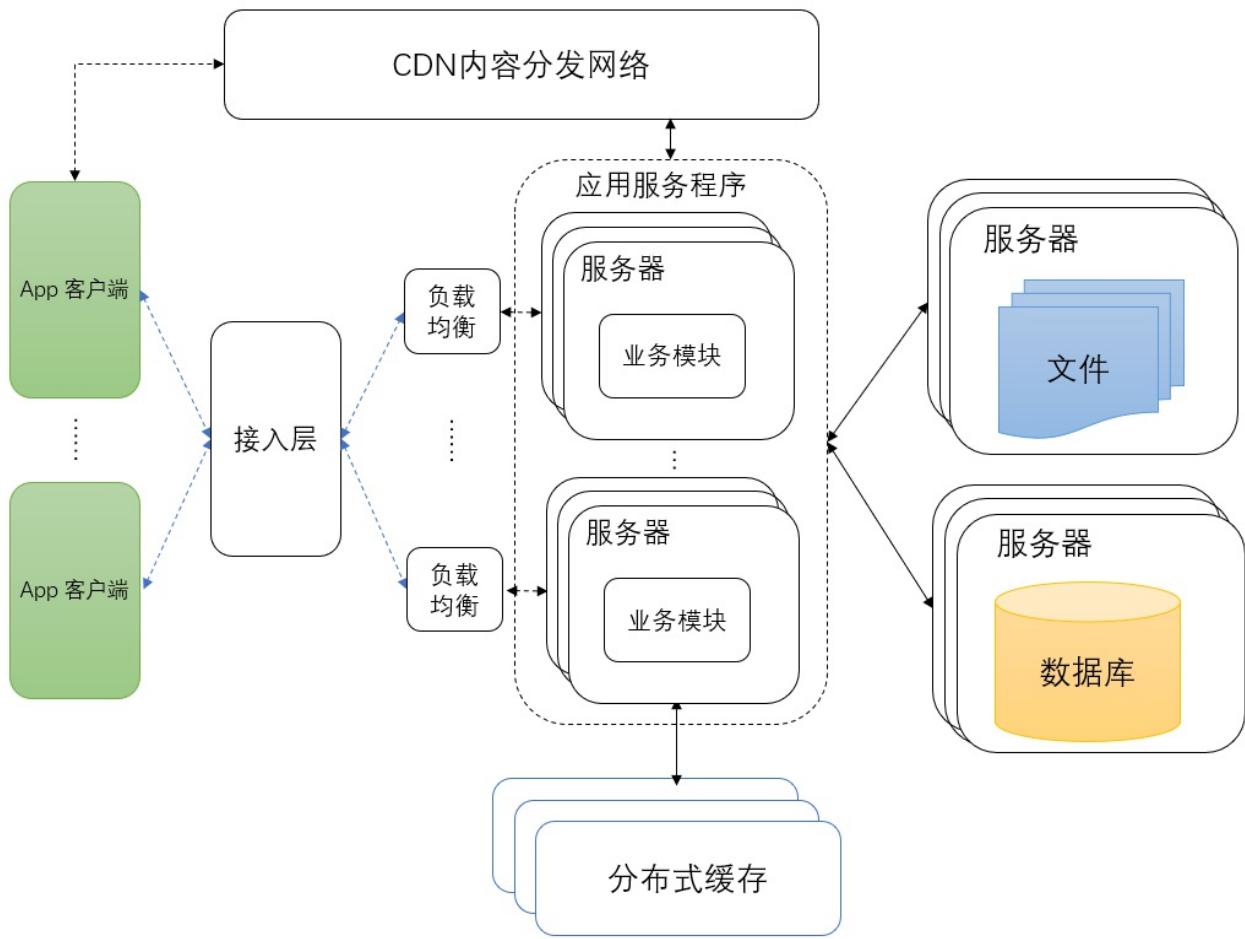
5. 分布式文件系统

随着用户数据的增长，产生的文件也越来越多，单台文件服务器已无法满足业务需要，需采用分布式文件系统以支撑。常见的分布式文件系统有 NFS。



6. 应用程序拆分

随着业务的发展，应用程序进一步膨胀，此时已不适合将其所有的组件部署在一起，而是需要按业务模块进行应用程序的拆分。每个业务模块负责相对独立的业务运作，包括版本迭代更新、业务演进。业务模块之间通过数据库或消息请求进行处理。现在主流的业务模块通信会使用 RESTful API 进行通信。各业务模块有自己的缓存系统、文件服务器系统和分布式数据库系统。



以上即为通用大型服务器方案架构演进路线。

小结

本小节简单高度概括了服务器端演进的整个过程。这里只提供一些思路，具体的架构和方案还要看具体的项目而行。后台架构方案可以千差万别，适合自己业务模式的才是最好的。

上面提到的很多技术，如负载均衡、CDN 分发、分布式缓存和分布式数据库等，在传统的架设中，从物理连线到软件安装，都需要人力投入。自从公有云服务推出后，这些技术已经作为公有云的基础设施推给客户。现在的企业，不用再自己去购买硬件设备并维护机房，而只需要通过购买服务的方式搭建这个业务生态环境，大大提高了工作效率及管理效率。