

如 MySpace 是一个过亿注册用户的超大型网站。在发展初期, MySpace 就是通过添加新的 web 服务器来提高性能的, 直到其数据库服务器过载。在第二代架构中, 启用了 3 个 SQL Server 数据库服务器, 一个为主, 接受所有的数据提交, 并将内容复制给另外两个, 由它们全力向用户提供数据。在这个架构中, 通过增加辅数据库服务器的方法, 可以有效应对系统访问量的增加。接着对数据库的架构进行了垂直分割, 不同的数据库服务器服务于不同类的功能, 有的负责登录, 有的负责博客等。当用户达到 200 万的时候, 启用存储区域网络(storage area network, SAN)。按 MySpaces 的说法, 此举极大提升了系统的性能、正常运行时间和可靠性。随着用户规模增加(300 万), 垂直分割不够用了, 毕竟有些信息(如用户表)需要在所有 DB 中共享, 维护数据一致性的开销很大。另外, 有一些应用增长太快, 其专用 DB 压力太大。这时候, 需要进行向上扩展(Scale Up: 升级服务器)和向外扩展(Scale Out: 加强分布式能力, 用大量便宜的服务器来分担压力)的抉择。为了尽量少改动以前的代码, 先考虑了向上扩展, 研究了如何使用 32CPU 的服务器的的问题。但最终, 还是走上了向外扩展的道路, 开始提炼分布式计算架构。这时候, 前面被拆分的应用在逻辑上又被整合了起来。并且, 用户以百万为一组, 被分别存储不同的 DB。再后来, 用户达到 1000 万时采用了新型的 SAN, 更改了 SAN 和数据库的绑定方式。在 Web 服务器和数据库服务器之间加上数据缓存层——这本是一开始就应该做的事情。

从 MySpace 的变化可以看出, 随着用户量的增加和用户数据量的积累, 数据库服务器将日益成为瓶颈。在它成为瓶颈之前, 充分做好各种相关的技术准备工作是必须的。