大型网站架构不得不考虑的 10 个问题

摘 要： 这里的大型网站架构只包括高互动性高交互性的数据型大型网站，基于大家众所周知的原因，我们就不谈新闻类和一些依靠 HTML 静态化就可以实现的架构了，我们以高负载高数据交换高数据流动性的网站为例，比如海内，开心网等类似的 web2.0 系列架构。....

这里的大型网站架构只包括高互动性高交互性的数据型大型网站，基于大家众所周知的原因，我们就不谈新闻类和一些依靠 HTML 静态化就可以实现的架构 了，我们以高负载高数据交换高数据流动性的网站为例，比如海内，开心网等类似的 web2.0 系列架构。我们这里不讨论是 PHP 还是 JSP 或者.NET 环 境，我们从架构的方面去看问题，实现语言方面并不是问题，语言的优势在于实现而不是好坏，不论你选择任何语言，架构都是必须要面对的。

这里讨论一下大型网站需要注意和考虑的问题

1、海量数据的处理

众所周知，对于一些相对小的站点来说，数据量并不是很大，select 和 update 就可以解决我们面对的问题，本身负载量不是很大，最多再加几个 索引就可以搞定。对于大型网站，每天的数据量可能就上百万，如果一个设计不好的多对多关系，在前期是没有任何问题的，但是随着用户的增长，数据量会是几何 级的增长的。在这个时候我们对于一个表的 select 和 update 的时候(还不说多表联合查询)的成本的非常高的。

2、数据并发的处理

在一些时候，2.0 的 CTO 都有个尚方宝剑，就是缓存。对于缓存，在高并发高处理的时候也是个大问题。在整个应用程序下，缓存是全局共享的，然而在 我们进行修改的时候就，如果两个或者多个请求同时对缓存有更新的要求的情况下，应用程序会直接的死掉。这个时候，就需要一个好的数据并发处理策略以及缓存 策略。

另外，就是数据库的死锁问题，也许平时我们感觉不到，死锁在高并发的情况下的出现的概率是非常高的，磁盘缓存就是一个大问题。

3、文件存贮的问题

对于一些支持文件上传的 2.0 的站点，在庆幸硬盘容量越来越大的时候我们更多的应该考虑的是文件应该如何被存储并且被有效的索引。常见的方案是对文 件按照日期和类型进行存贮。但是当文件量是海量的数据的情况下，如果一块硬盘存贮了 500 个 G 的琐碎文件，那么维护的时候和使用的时候磁盘的 Io 就是一个 巨大的问题，哪怕你的带宽足够，但是你的磁盘也未必响应过来。如果这个时候还涉及上传，磁盘很容易就 over 了。

也许用 raid 和专用存贮服务器能解决眼下的问题，但是还有个问题就是各地的访问问题，也许我们的服务器在北京，可能在云南或者新疆的访问速度如何解决?如果做分布式，那么我们的文件索引以及架构该如何规划。

所以我们不得不承认，文件存贮是个很不容易的问题

4、数据关系的处理

我们可以很容易的规划出一个符合第三范式的数据库，里面布满了多对多关系，还能用 GUID 来替换 INDENTIFY COLUMN 但是，多对多关系充斥的 2.0 时代，第三范式是第一个应该被抛弃的。必须有效的把多表联合查询降到最低。

5、数据索引的问题

众所周知，索引是提高数据库效率查询的最方面最廉价最容易实现的方案。但是，在高 UPDATE 的情况下，update 和 delete 付出的成本会高的无法想想，笔者遇到过一个情况，在更新一个聚焦索引的时候需要 10 分钟来完成，那么对于站点来说，这些基本上是不可忍受的。

索引和更新是一对天生的冤家，问题 A，D，E 这些是我们在做架构的时候不得不考虑的问题，并且也可能是花费时间最多的问题。

6、分布式处理

对于 2.0 网站由于其高互动性，CDN 实现的效果基本上为 0，内容是实时更新的，我们常规的处理。为了保证各地的访问速度，我们就需要面对一个绝大的问题，就是如何有效的实现数据同步和更新，实现各地服务器的实时通讯有是一个不得不需要考虑的问题。

7、Ajax 的利弊分析

成也 AJAX，败也 AJAX，AJAX 成为了主流趋势，突然发现基于 XMLHTTP 的 post 和 get 是如此的容易。客户端 get 或者 post 到服务器数据，服务器接到数据请求之后返回来，这是一个很正常的 AJAX 请求。但是在 AJAX 处理的时候，如果我们使用一个抓包工具的话，对数据返回和处 理是一目了然。对于一些计算量大的 AJAX 请求的话，我们可以构造一个发包机，很容易就可以把一个 webserver 干掉。

8、数据安全性的分析

对于 HTTP 协议来说，数据包都是明文传输的，也许我们可以说我们可以用加密啊，但是对于 G 问题来说的话，加密的过程就可能是明文了(比如我们知道 的 QQ，可以很容易的判断他的加密，并有效的写一个跟他一样的加密和解密方法出来的)。当你站点流量不是很大的时候没有人会在乎你，但是当你流量上来之 后，那么所谓的外挂，所谓的群发就会接踵而来(从 qq 一开始的群发可见端倪)。也许我们可以很的意的说，我们可以采用更高级别的判断甚至 HTTPS 来实 现，注意，当你做这些处理的时候付出的将是海量的

database，io 以及 CPU 的成本。对于一些群发，基本上是不可能的。笔者已经可以实现对于百度空 间和 qq 空间的群发了。大家愿意试试，实际上并不是很难。

9、数据同步和集群的处理的问题

当我们的一台 databaseserver 不堪重负的时候，这个时候我们就需要做基于数据库的负载和集群了。而这个时候可能是最让人困扰的的问题 了，数据基于网络传输根据数据库的设计的不同，数据延迟是很可怕的问题，也是不可避免的问题，这样的话，我们就需要通过另外的手段来保证在这延迟的几秒或 者更长的几分钟时间内，实现有效的交互。比如数据散列，分割，内容处理等等问题。

10、数据共享的渠道以及 OPENAPI 趋势

Openapi 已经成为一个不可避免的趋势，从 google，facebook，myspace 到海内校内，都在考虑这个问题，它可以更有效的留住 用户并激发用户的更多的兴趣以及让更多的人帮助你做最有效的开发。这个时候一个有效的数据共享平台，数据开放平台就成为必不可少的途径了，而在开放的接口 的情况保证数据的安全性和性能，又是一个我们必须要认真思考的问题了。