

疯狂代码,大型网站架构系列之二，底层架构

概论

疯狂代码 <http://www.CrazyCoder.cn/> j:<http://www.CrazyCoder.cn/Yuanchuang/Article11064.html>

书结上回，www.crazycoder.cn/Yuanchuang/Article10919.html ;首先澄清上篇中关于几个朋友的评论。上篇疯狂代码介绍的基于AJAX的攻击很多人提出疑问，比如不能跨域，减轻负担之类。Ajax是通过简单的GET和POST进行数据传递的，采用HTTPDEBUGGER，抓取数据，然后采用如下方案，顺便写个示例的攻击代码。比传统的webform，我们更容易构造一些，其实对于webform和ajax的处理和发包过程是一样的，ajax数据量相对小，速度也快一些。结合SharpPcap和HttpWebRequest我们构造一个合理的正常的IP数据包过去,代码很长，我们用伪代码简单的表达一下。`request.CreateUrl(Ajax处理页面);`
`request.Method=" GetORPost" ;request.refere=" 网页来源" ;SharpPcap.SetLinkConnection(伪造IP地址);String content = request.GetResponseStream()` 如果作为一个多线程的应用程序对对方的WEB构成批量发包的话（假如是DEDECMS），足可以把Dedecms的数据库搞垮;文入正题：对于上回书提到要解决问题A，我们先讲解一下电信公司ADSL的布局方案。机器上没有装VISIO，我简单的用文字描述一下流程。Adsl用户A输入用户名密码远程连接到账户数据库（在天津）账户数据库连接计费数据库并返回状态如果成功，连接PPPOE服务器，并进一步连接计费数据库认证服务并连接。;这里没有什么特别的地方，但是和qq通讯服务是一样的，就是采用了统一的用户验证服务器，同时对于用户验证的信息数据库是只读的，我们从其中可以想到什么吗？;以上是个简单的例子，下面开始谈具体的架构策略，首先对于上篇提到的问题A，我们首先以用户数据库为例来做解释和要求。首先做用户量估算需求，假如我们做的是学术社区，那么这个用户量不会很大，可能我们不需要考虑这个，对于用户量的级别，我们暂时把用户量级别定为三种，百万级别（M）和千万级别（S），以及亿万级别（Q），并考虑用户登录验证以及查询常用的操作，对M和S进行扩充以及了解。;众所周知，在这个情况下，对于用户数据的负载其实并非可行而不可行的问题，而是如何最大化的保证查询和更新以及各个服务器之间的数据同步。这里我们不再讲解如何优化如何索引，只介绍架构初期的方案，下面介绍的方案如果涉及全表查询，可以采用分区视图的方案，大家可以具体搜索相关资料。;对于M级别来说，现有的DBMS经过合理的布局完全可以满足需求。我们需要解决的问题的关键其实是处理IO方面的问题，处理方案相对简单一些，对数据库的FILE文件分磁盘存贮（不是分区，是不同的硬盘），根据负载量大小，我们可以适当的控制硬盘的数量和文件分区的数量。;对于S级别，上个处理方案已经不能完全满足需求了，这个时候我们需要对注册和入库的流程进行简单的修改了，解决方案是数据散列和分区视图的概念，具体概念大家去google一下，我不详细说明了。我们常用的方案有三种。第一种是等容扩充法，在用户注册控制的基础上，保证每个库的用户容量不超过500万，超过之后入第二个库，以此类推，这个方案可以保证系统有效的扩充性，但不能保证数据被有效的索引。第二种就是共区索引方案，其实和第一种方案有异曲同工之说但是讲第一种方案进行了合理的优化，按照用户名进行分库存贮。比如我们可以建立26的数据库，按照用户名的索引来控制用户数据入哪个库。假如用户名是CrazyCoder，那么就讲该用户名的数据存放在用户表C中，在数据存贮的时候可以很方便的根据用户名进行相应的数据查询，方案二可以有效的解决数据索引问题。方案三是一个更具模型化的方案，结合方案一和方案二，进行用户ID的编码，不是IDENTIFY Cloumn，我们用一种序列化的方案将用户名以编码的形式存贮，比如用户名是CrazyCoder,我们的编码方案就是通过算法进行数字化，将CrazyCoder按照

C, R, A,存贮为数字索引, 然后进行分区存贮, 数字类型的数据在数据库中可以更有效的被查询和被更新和共享, 结合方案一和方案二这个就是方案三。; ;对于Q级别。数据量已经是足够海量了, 这个时候无论用哪种方案都是一个让人头大的数据, 不能简单的用查询的方案来处理了, 可以参考S级别的进行处理。但这个时候我们采用的方案是根据用户活跃度的权值结合数据量进行临时数据表的存放。如果一个非意外的数据情况下, 每天登录的用户量不会上千万。这个时候我们需要做的是简单的数据代理程序。一个临时的用户验证数据库, 每天执行一次批处理, 将活跃度高的用户账户提取到临时数据库中, 查询的时候先查询临时库, 如果没有在进行全库查询。这个根据系统的负载情况来估计阈值, 不同的系统估算方案也不尽相同。; ;上面对于 M, S, Q三种级别进行了简单的概述, 下面介绍一个在其之上更高级的一个查询方案, 数据缓存服务器, 我们也可以把它理解为缓冲服务器, 数据做为只读来使用。; 具体实现方案如下: 以为涉及了海量, D B M S 常规的缓存方案已经不符合我们的要求了, 那么我们需要一个有效的缓存方案, 这个时候处理的流程其实就是讲最常用最直接的数据直接存放在缓存服务器中, 而这个缓存服务器定时从主服务器获取并更新信息。这个是一个简单的查询, 我们还可以更深入的讲缓存服务器做二次缓存, 也就是一次性处理输入并存放于 L I S T 数据中, 作为全局变量放到内存中进行查询, 同时用 H A S H T A B L E 或者数组进行数据组索引 (可以是多级), 根据查询分布到各个变量中。直接从内存中读取数据。; 以笔者的经验来说的话, 对于 I T E M 数据不超过 1 0 K 来说, 每个列表最佳的存放范围是 0 到 6 万之间。; 这里简单的介绍了一下 DBMS 基本架构, 里面具体细节处理的还有很多, 这里只介绍个大概的纲要。有问题请给我发邮件(Heroqst ;# ;Gmail.com), 请讲#替换为@; ;这里只是简单的介绍了一下 D B M S 的基本布局, 下章讲具体对我们常见的多对多关系数据库进行具体配置说明。; 首先介绍一下问题的大概, 比如对于文章和标签, 每个文章可以有多个标签, 而每个标签下又会有多个文章, 那么数据量将是文章数乘以标签数, 这个时候如何进行处理并有效的索引, 将是下章要介绍的内容。; 疯狂代码, 大型网站架构系列, 同步发布于(<http://www.admin5.com>和<http://www.crazycoder.cn>), 转载请注明出处。; 2008-9-17 10:30:54

疯狂代码 <http://www.CrazyCoder.cn/>