从 XToolsCRM 谈构建安全 SaaS 构架

■ 文 / 李亚平

很多程序员简单认为：将任何一套BS软件改装，放在公网服务器上，就可以成为一个SaaS服务型产品进行销售。这种现象导致很多SaaS运营业务搁浅，究竟是什么原因？在成功的运营XToolsCRM四年后，我认为：根不正则叶不茂；做SaaS不仅需要做好应用级产品，更主要的是做好SaaS的底层架构。

概述

一套合格的SaaS产品，除了具有优秀的应用及产品外，还需要一套合理的基础构架，这些基础构架包含：可以随意扩充的服务器集群架构、数据必须有灾备设施；安全的集群结构，必须保证客户和客户之间数据的隔离；运营服务器上不能放程序源码，版本控制和程序升级体系……

基础集群概念

首先，我们讨论一下设计容量问题，因为下面所有的构架都建立在设计容量这个参数上，从XToolsCRM角度上，我们将设计容量定义为：每台服务器设计支撑2000家企业用户，其中活动用户500~800家企业。

为什么分支撑用户和活动用户呢？按照XToolsCRM的经验，用户需要试用才能租用，在试用期结束后，XToolsCRM会给客户一个数据保留期，在保留期内续费数据就不会丢失。但是这样，大量的客户数据会保存在服务器上，对服务器的物理结构和逻辑结构近乎苛刻。

在用户大幅度扩展后，我们只需要花数个小时，就能将新的服务器上线并网运行，让客户的数据和服务器的数量呈线性关系，不会出现瓶颈。

所谓的服务器集群，就是分为管理服务器和应用服务器两个部分，也就是说帐号的开设、付费、续费等等管理操作都在管理服务器上，而客户使用的软件都在应用服务器上。

由于中国的网络特点，我们把服务器分成四个区域部署，分别是：网通机房（针对长江以北）、电信机房（针对长江以南）、多路机房（针对全国都开有分公司的公司）、异地容灾备份中心（备用）；在实际应用中，为了让用户得到更快捷的网络服务，我们还在各大城市直接部署服务器。在每个地区的服务器集群中，由于Web处理的负载程度，可以根据实际情况配置应用服务器的数量。

另外，整个系统在设计上是分割成几套独立的子系统，即便是在管理服务器损坏的情况下，各个子系统都可以独立运行。

服务器的安全措施

安全分为三个部分：对外安全，容灾备份，数据隔离。

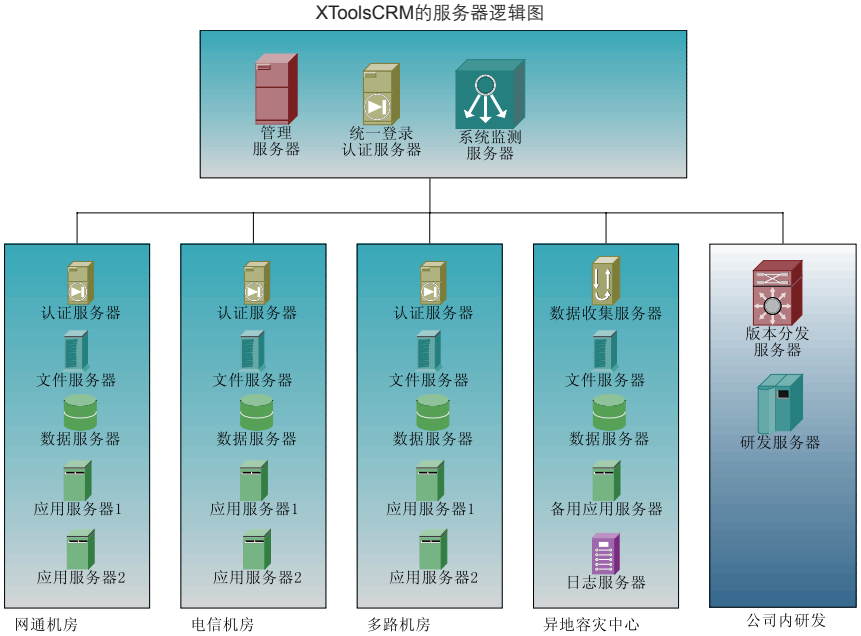
对外安全

我认为对外安全主要应该做到：

1、全面采用linux服务器，并且剔除与CRM无关的服务及软件系统；

2、采用全球数据安全证书，使用

XToolsCRM的服务器逻辑图



SSL加密的https协议。

3、所有运营服务器不允许程序员调试，运营服务器上不允许有程序源码，而所有的调试都在研发服务器上进行，经过测试、代码审查后，封装成版本，由版本分发服务器升级到各个运营服务器。

4、防范SQL注入，注重代码防范、SQL监控、安全码技术等综合防范。

5、处理好数据、代码之间的关系，数据区和代码区完全隔离。

容灾备份

在整个系统中，因为服务器分布在全国各地，如果出现意外事故，如果按照常规重新装机，恢复时间也需数天，在SaaS服务中这是不可取的。

在容灾备份中心，我们使用了一个“数据收集”的概念，在每台数据库执行SQL变更语句时，系统都会将变更的SQL语句记录下载，通过“数据收集”汇总在容灾中心的数据服务器上，并且做到了一对多备份；该备份最小间隔为3分钟。

对于文件来说，考虑到CRM的实际需求，文件暂时使用每天增量备份传输。

CRM使用会产生庞大的日志，我们需要专门的日志服务器来管理这些日志，已方便用户查询相应的操作记录。

备用应用服务器可以在特殊的时候切换成应用服务器来替换任意一台出问题的服务器，

数据隔离

由于SaaS是多家企业使用一套系统，在服务器内部，如何做到企业间数据的隔离呢？最好的办法当然是在操作系统外使用VM软件（虚拟机）可以做到真正的隔离，但是这样成本很高，每台机器大约只能提供5个用户，这种模式无法满足SaaS的应用。

XToolsCRM认为，良好的企业间的数据隔离应该基于服务器模式，明确每个逻辑服务器的作用是很有必要的。

认证服务器：主要处理用户身份确认和生成动态密码的作用，内部访问接口简单且安全。

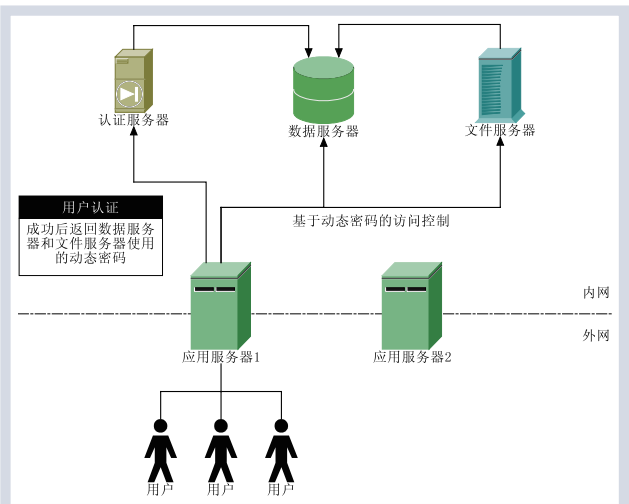
应用服务器：全部的应用程序，但是没有任何数据存储，可以直接替换，甚至可以使用SSD小容量硬盘。

数据服务器：一个企业使用一个独立的DATABASE，并且每个DATABASE拥有独立的密码，并且改密码可以动态修改。

文件服务器：用户上传的任何东西都只能列为数据文件，为了安全起见，需要改名存储到特定位置，决不能简单的存储到web服务器的wwwroot之内，根据需求，我们制定了list，up，down三个简单的API。

按照上述模式，由于应用服务器的全部代码都是编译过的且处于ROM状态，因此想通过一个企业帐号获取另外一个企业帐号数据的可能性基本不存在。

如果数据库使用简单的单DATABASE、靠字段做企业分离这是根本不可取的，任何一个低级错误（比如URL修改id、查询条件人为注入:` or 1 ）等都有可能泄露数据，这种系统对企业来说简直就是恶梦。



执行效率的问题

SaaS模式必须考虑各逻辑服务器的执行效率，效率越高意味着单用户成本越低，市场竞争力越高，实际上来说效率在应用服务器、数据服务器、文件服务器上都要注意。

应用服务器：大量的客户的请求、页面合成等操作都基于应用服务器，因此该服务器主要瓶颈是CPU，应使用多核处理器，且代码进行编译和优化。

数据服务器：按照数据库经验来说，使用单DATABASE,字段做企业分离这种模式效率会高很多，尤其是大内存模式，但是由于安全需要，我们不得不选择更安全的独立DATABASE模式，而且这个模式会让用户在迁移服务器时更便捷，但是带来的后果就是服务器必须同时打开更多的table，这需要我们针对服务器操作系统做特殊的优化才能达到我们的要求。

文件服务器：也许有人并不认为文件服务器很重要，甚至认为很简单，其实不然，大量的小文件存储导致文件碎片急剧增加，严重影响效率(尤其是Windows服务器)，而对于Linux来说，这也是很致命的，需要独立的文件存储区域、碎片整理机制和缓存机制。按照设计容量来说，5台应用服务器*2000企业*10000文件*10%(使用率)=1千万个文件；想象一下，即便是Windows的C盘，也就是5~6万个文件，处理不当就会感觉硬盘缓慢，那么1千万个文件是什么概念呢？解决办法有很多，比如：独立分区、将文件自动区分为活跃文件和长期存储文件两种区域单独处理。

作者简介

李亚平，北京沃力森信息技术有限公司CTO，20余年编程经验，早年间推出DOS下的畅销软件VTTE（可视磁盘刊物生成器），参与“英汉通”等词典软件的开发；其后从事多年的软件构架设计。2004年创立XToolsCRM。