

一种虚拟化的服务器集群模型^①

崔荣波, 许晓兵, 陆颖

(上海理工大学 管理学院, 上海 200093)

摘要: 随着经济与电子商务的高速发展, 企业与用户的业务交互次数呈现出几何级数增长的趋势, 然而企业服务器的利用率却只有 15%~30%。阐述了在 Linux 平台上运用虚拟化技术将企业现有的服务器虚拟成多台服务器, 同时运用负载均衡调度算法对虚拟的多台 Linux 客户机进行集群的方案模型, 以达到强化企业业务的持续性, 加固数据的完整性和安全性、降低 IT 部门在整个企业范围内的成本的目的。

关键词: 虚拟化; 服务器集群; 负载平衡; Linux 平台

Server Clustering on Virtualization

CUI Rong-Bo, XU Xiao-Bing, LU Ying

(Business School, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: With the rapid development of the economic and electronic commerce, the exchanging times between the enterprises and users is increasing as geometric series. However, the utilization rate of enterprise servers is only 15% to 30%. This papers illustrates a program about the combination of the Linux platform and virtualization technology. The existing server is transferred into multiple virtual servers with the VMware Workstation. At the same time, the virtual Linux clients are clustered using the VM virtualization technology and scheduling algorithm for load balancing. With this method, we cannot only strengthen its business continuity and the security and integrity of the data but also help the IT departments reduce the whole costs.

Keywords: virtualization; server clustering; load balancing; linux platform

1 引言

随着计算机技术和 Internet 服务的发展, 企业与用户之间的业务数据交互已经基本转移到互联网平台之上。经济与电子商务的快速发展使得用户与企业通过网络进行业务交互的次数呈现出几何级数增长的趋势, 这对企业服务器的高可用性和高扩展性提出了很高的要求。因为任何服务器中断都会对企业造成不可估量的损失, 如何在不增加企业 IT 成本的前提下, 保证企业业务的持续性, 成为企业 IT 部门必须考虑的问题。显然传统的单服务器已经不能满足需求, 而购买多台服务器又会大大增加企业的 IT 成本。

由于 IBM、微软、EMC 等大公司的鼎力支持, 虚

拟化技术得到飞速发展, 使得这个问题迎刃而解。应用这种技术不但强化业务的持续性, 加固数据的安全性, 同时帮助企业降低信息化的总成本。

2 模型中使用的技术与算法

2.1 虚拟化技术

虚拟化意指资源的打包和分离, 允许企业在单一接口上管理不同的资源, 降低管理的复杂度及成本。这种技术实现了在同一实体系统上运行多个操作系统, 并允许系统资源在各个操作系统间按需求来进行资源分配。

虚拟化技术的提出是因为人们发现在生产环境

^① 基金项目: 高等学校博士点基金(20093120110008); 国家自然科学基金(70903047); 上海市重点学科建设项目(S30504)

收稿时间: 2010-07-07; 收到修改稿时间: 2010-08-30

中，服务器的实际负载率很低，企业服务器仅有 15%~30% 的系统处理能力，绝大部分的服务器负载都低于 40%，使得 IT 投资回报率偏低。在这种情况下，可以利用虚拟化技术，在一个物理服务器设备之上建立若干虚拟服务器系统，使之能够动态的分配资源，最终提高服务器的实际负载率，从而可以降低 IT 投资成本。虚拟化结构如图 1 所示。

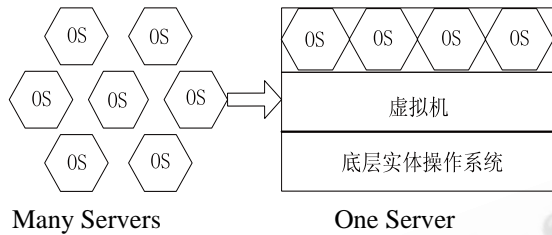


图 1 虚拟化结构示意图

实现 Linux 虚拟化的方法有多种，它们都可以通过不同层次的抽象来实现相同的功能，常用的四种虚拟化方法^[1]：硬件仿真、完全虚拟化、准虚拟化以及操作系统层虚拟化。

2.2 IP 负载均衡技术

在负载均衡调度器的实现技术中，IP 负载均衡技术起到至关重要的作用。在已有的 IP 负载均衡技术中有通过网络地址转换将一组服务器构成一个高性能的、高可用的虚拟服务器，我们称之为 VS/NAT(Virtual Server via Network Address Translation)技术，大多数商品化的 IP 负载均衡调度器产品都是使用此方法。在分析 VS/NAT 的缺点和网络服务的非对称性的基础上，我们提出通过 IP 隧道实现虚拟服务器的方法 VS/TUN(Virtual Server via IP Tunneling)，和通过直接路由实现虚拟服务器的方法 VS/DR(Virtual Server via Direct Routing)，它们可以极大地提高系统的伸缩性。

以上三种 IP 负载均衡技术的优缺点归纳在表 1 中：

表 1 负载均衡技术的优缺点

	VS/NAT	VS/TUN	VS/DR
Server	any	Tunneling	Non-arp device
server network	private	LAN/WAN	LAN
server number	low (10~20)	High (100)	High (100)
server gateway	load balancer	own router	Own router

以上三种方法分别所能支持最大服务器数目是依

照以下条件下进行估计的，1)假设调度器使用 100M 网卡，2)调度器的硬件配置与后端服务器的硬件配置相同，而且是对一般 Web 服务。如使用更高的硬件配置(如千兆网卡和更快的处理器)作为调度器，调度器所能调度的服务器数量会相应增加。

2.3 调度算法

负载均衡调度是以连接为粒度的。按照 HTTP 协议，从 WEB 服务器上获取每个对象都需要建立一个 TCP 连接，而且同一用户的不同请求会被调度到不同的服务器上。因而这种细粒度的调度完全避免了用户访问的突发性引起的负载不平衡。下面介绍三种已实现的调度算法^[2]：

(1) 最小连接调度，它需要记录各个服务器已建立 TCP 连接的数目，把新的连接请求发送到当前连接数最小的服务器。当各个服务器有相同的处理性能时，最小连接调度能把负载变化大的请求平滑分布到各个服务器上，所有处理时间比较长的请求不可能被发送到同一台服务器上。

(2) 加权最小连接调度，当各个服务器的处理能力不同时，最小连接调度算法并不理想，因为 TCP 连接处理请求后会进入 TIME_WAIT 状态，加权最小连接调度是最小连接调度的超集，各个服务器用相应的权值表示其处理性能。

假设每台服务器的权值为： $W_i (i = 1 \dots n)$

TCP 连接数目： $T_i (i = 1 \dots n)$

依次选： T_i / W_i

为最小者的服务器作为调度对象。

(3) 持续的客户端连接调度算法，所有来自同一个 IP 的客户端将一直连接到同一个物理服务器，直到超过限时。Pcc 是为 HTTP 和 COOKIE 服务设置的。在这种调度规则下，第一次连接后，所有以后来自相同客户端的连接(包括来自其它端口)将会发送到相同的物理服务器。

3 模型的建立与分析

建立模型，假设企业开始有两台服务器，一台正在使用，另一台只做定期备份，以备发生故障时接入，以保证公司业务的持续性。按照这种方法运行，在更换服务器时，可能对企业造成重大损失(因为在更换时，服务器是间断的)。

同时,如果一台服务器只装一个操作系统,它的最大利用率不超过 40%。利用虚拟化技术,可将一台服务器变成多台虚拟服务器。根据这个思想,我们将企业的一台服务器虚拟成 WEB 服务器和数据库服务器两台服务器,并分别为每台虚拟服务器虚拟三台客户机,同时使用磁盘阵列技术^[3]来保证数据的完整性。其虚拟体系结构如图 2 所示。

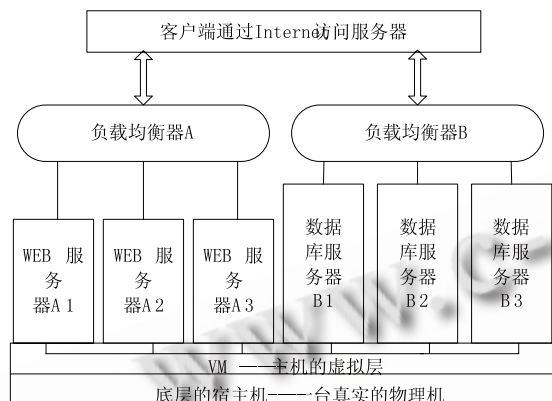


图 2 模型中的虚拟体系结构

3.1 底层宿主机的硬件要求与 OS 的选择

底层主机上要运行多个虚拟的操作系统,系统负荷重。因此需要从 CPU、内存及 I/O 三个方面配置底层平台^[4]。

操作系统作为宿主机的支撑平台,需要从系统的稳定性、运行效益以及客户机操作系统几方面综合考虑。由于 Linux 平台在内存分配、系统资源占用,进程调度、远程管理等方面比 Windows 更加高效与灵活,并且在易用性与管理性方面具有优势。故该模型采用 Red hat Enterprise Linux5 作为宿主机的系统平台。

如果虚拟机与宿主机为异种操作系统时,不同文件格式的转化带来较大的系统开销,降低了虚拟机的 I/O 性能。所以模型中的虚拟机同样采用 Linux 操作系统。

3.2 构建模型中的虚拟网络结构

模型中将一台宿主机虚拟成六台 Linux 映像客户机^[5],同时为每台客户机虚拟一个或多个网卡,客户机操作系统与外界交流的数据都是通过底层的宿主机的网卡进行中转的。每台虚拟机拥有一个或者几个不同网段的 IP 地址,因此它们可以构建若干个客户机局

域网,每个客户机局域网关联一个虚拟网段,在此虚拟网段上可以定义若干网络接口卡(NIC),类似于网络接口,用来关联每个客户机局域网上的主机。同时针对不同的硬件接口设备有不同的虚拟设备相对应,如图 3 所示。

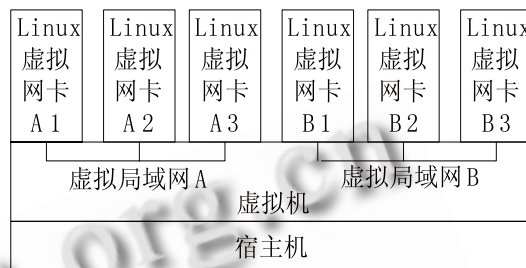


图 3 虚拟网络结构

3.3 模型中三种技术的实验分析

下面针对模型中所提出的四种虚拟化方法、三种 IP 负载均衡技术和三种调度算法的实验结果进行分析介绍。最后给出本服务器模型所取得效能与传统服务器模型的效能进行量化的分析和比较。

3.3.1 四种虚拟化技术的实验结果分析

对于四种虚拟化技术我们主要从 CPU 使用的角度进行量化比较。使用服务器压力测试工具 Load Runner 模拟大量用户同时访问服务器,记录底层宿主机的 CPU 使用率,如图 4 所示。由于硬件仿真是在宿主系统上创建一个硬件虚拟机来仿真所想要的硬件,每条指令都必须在底层硬件上进行仿真,因此对 CPU 的占用率很高,且随着用户访问的次数的激增,整个服务器可能会崩溃。再看完全虚拟化,该方法使用虚拟软件,在虚拟服务器和底层硬件之间建立一个抽象层,当然利用抽象层进行代码转换会带来一定的系统开销,但是随着 CPU 辅助虚拟化硬件技术的出现,完全虚拟化由软件层模拟的功能已逐步转向硬件化,虚拟化效率不断在提高,因此大幅降低了 CPU 的占用率。准虚拟化,通过使用虚拟化应用接口来定制客户机操作系统内核,以替代不可虚拟的某些特权指令,因此当大量用户涉及到某些特权指令时, CPU 的占用率会迅速上升。操作系统层虚拟化,没有独立的虚拟层,主机操作系统本身负责在多个虚拟服务器之间分配硬件资源,大量用户访问时,仅主机的分配动作就会占用大量资源。

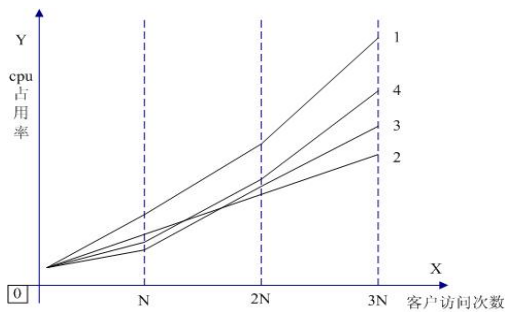


图 4 CPU 占用率比较

(注释：1 代表硬件仿真，2 代表完全虚拟化，3 代表准虚拟化，4 代表操作系统层虚拟化)

经过对实验结果的分析，该服务器模型如果采用完全虚拟化技术，可以大大降低 CPU 的使用率，服务器具有较高的性能。典型代表有 VMware^[6]等相关软件。

3.3.2 三种 IP 负载均衡技术的实验结果分析

IP 负载均衡技术的选择对于负载均衡器的效率非常重要。现假设有三台服务器，集群在一个负载均衡器上，有三个客户端，在每个客户端上都安装 Load Runner 压力测试程序，同时启动 IIS 服务器，三台服务器通过 cisco catalyst 2900 x1 交换机和主节点服务器(Load Balancer)相连，三台客户机通过 3com 3300 交换机和主节点服务器相连。在此服务器集群体系中为每个集群节点配置网络负载均衡服务，并分别采用三种 IP 技术进行仿真实验，其拓扑结构如图 5 所示。

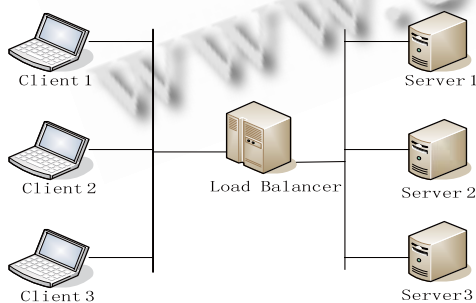


图 5 集群系统架构

(1) 性能测试指数

吞吐量=负载程序数/总耗时时间=负载程序数/(负

载程序数*平均查询时间*0.8+负载程序数*平均写入时间*0.2)

平均执行时间=平均查询时间*0.8+平均写入实际*0.2

其中，平均执行时间这一项指标是性能测试指标的综合反映。

(2) 测试结果分析,如下表所示:

表 2 平均执行时间性能对比表

B \ C	100	200	500	1000
A \ 网络地址转换	7.3	11.2	14.3	17.8
直接路由	6.8	10.1	12.9	15.8
IP 隧道	6.6	10.9	13.7	16.9

(注：A 代表 IP 技术，B 代表平均执行时间(单位：毫秒)，C 代表同时访问的用户数)

经过分析，三台服务器组成的集群系统，如果大量客户并发访问，采用 IP 隧道技术(VS/TUN)实现虚拟服务器，可以在某种程度上提高系统的整体运行效率。相同负载下的集群系统的应用程序的平均执行时间将会减少，从而提高集群系统的吞吐量。

3.3.3 三种连接调度算法的实验结果分析

调度算法的选择同样对负载均衡器的效率非常重要。建立与图 5 相同拓扑结构的模型，分别采用最小连接调度算法、加权最小连接调度算法和持续的客户端连接调度算法进行试验。

下面以数据库集群来测试三种调度算法，模型部署如下，选择 ORACLE11g 数据库，将其分别安装在图 5 的三个服务器上，在 Load Balancer 上安装 oracle 自己的数据库集群软件 clusterware，实现真正的应用程序集群（即 RAC:REAL APPLICATION CLUSTER）。

当数据库的并发访问数较小时，三种算法的性能差不多；当访问量逐渐增多时，又分为两种情形，一种是分布式，在这种情形下，持续的客户端连接调度算法呈现出自己的优势，它会快速的将请求发送到最近区域的分布式服务器上，并且可以自定义超时时间。第二种情形是非分布式，在访问并发量激增的过程中，加权最小连接调度算法较最小连接调度算法的优势逐

渐显现出来, 因为 TCP 链接达到一定数量时, 最小连接调度算法会使服务器进入等待状态, 而加权最小连接调度是最小连接调度的超集, 它可以使用加权算法找到最小链接的服务器, 最大限度的降低访问的等待时间。

综上三种实验所述, 本服务器模型比传统服务器模型在效能上有很大的优势。具体表现在: 它可以使企业服务器的利用率从传统的 15%~30% 提高到 80% 以上; 高强度的企业业务的持续性; 高可靠的数据完整性和安全性, 以及低廉的设备成本和维护成本等。

4 应用实例

经过第三部分的分析, 该实例将采用 Red hat Enterprise Linux5 构建宿主机的系统平台; 采用 VMware Server2.0 实现完全虚拟化技术, 虚拟多个客户机操作系统; 采用 Apache+Tomcat+JK 实现 WEB 端的集群配置; 采用甲骨文公司的 RAC(Real Application Cluster)实现对 oracle10g 数据库, 的集群配置。

4.1 底层宿主机的构建

在服务器领域, Linux 拥有 90% 以上的市场占有率。笔者将选择 Red hat Enterprise Linux5 构建底层平台, 它可以从 <http://www.verycd.com/topics/2719336/> 网站上下载。为了实现更高的数据安全, 建议将 Linux 系统分为 8 个分区, 即:

- root 分区: 用 “/” 表示, 大概 100-500M, 主要用于存放系统配置文件、服务程序等。

- 交换分区: 用 “/swap” 表示, 一般为内存的 2 倍大小, 被用作系统的虚拟内存。

- 内核分区: 用 “/boot” 表示, 用于存放 Linux 的 kernel。

- 用户安装程序的分区: 用 “/usr” 表示。

- 用户个人数据分区: 用 “/home” 表示。

- 存储临时文件的分区: 用 “/var” 表示。

- 存放第三方软件的分区: 用 “/opt” 表示。

这样分区有很多优点, 例如: 可以实现 Linux 系统的最小重装; 保证用户个人数据的安全; 防止恶意

攻击, 造成系统临时文件大量堆栈, 使系统崩溃等问题。

4.2 虚拟层的构建

完全虚拟化技术的典型代表是 VMware 公司的 VMware Server, 当前发布版本为 V2.0, 它有 Windows 和 Linux 两个发行版本。这两个版本都可从 <http://www.vmware.com/> 网站上获取, 在线注册后免费获取授权码。

首先安装 VMware Server2.0, 创建系统虚拟层, 然后在该层上创建一个客户机操作系统, 并使用虚拟机的克隆(Clone)技术快速创建其余 5 个客户机系统。另外, 我们还可以使用 VMware Converter 工具将现有的服务器上的资源迁移到虚拟机中, 使客户机的创建更加简便。接下来将这 6 台客户机分成两组, 一组做 WEB 集群使用(A1、A2、A3), 另外一组做数据库集群使用(B1、B2、B3)。

4.3 集群的构建

4.3.1 WEB 集群的配置

WEB 集群将采用 Apache+Tomcat+JK 进行集群配置, JK 是 Tomcat-Apache 插件, 为 Apache 和 Tomcat 的连接器, 处理 Tomcat 和 Apache 之间的通信, 在集群配置中充当负载均衡器的作用。具体配置步骤如下:

- 1) 在图 2 的 A1 上安装 1 个 apache 和 1 个 tomcat, 在 A2 和 A3 上分别安装一个 tomcat。

- 2) 配置 Apache。复制插件 jk_module 到 apache 的 modules 目录中。建议采用 JK1.2, 不要用 JK2(由于其配置太过麻烦, 使用的人很少, 所以已停止开发)。

- 3) 在 Httpd.conf 中配置 JK。即: 在 Apache 安装目录下的 conf 子目录中, 打开 httpd.conf 配置文件, 在最后插入以下一行: Include conf/mod_jk.conf

- 4) 配置 httpd.conf 的虚拟主机部分, 将需要 tomcat 处理的 httpRequet 交给 tomcat。

- 5) 配置 JK 需要的 workers.properties 文件。

4.3.2 数据库集群的配置

该实例使用 oracle10g 数据库, 并采用甲骨文公司

(下转第 213 页)

getshahash 获取参数 psw 的密码摘要。

要验证身份是否正确，需要获得被判断密码的 Hash 值，然后与保存的密码摘要进行比较，如果完全相同，则通过身份验证。这样设计的目的是，即使密码被人非法窃取，只要不是在本机上使用软件，他就没办法获取注册表中的信息，从而保障合法用户的权限。身份验证流程如图 3 所示：

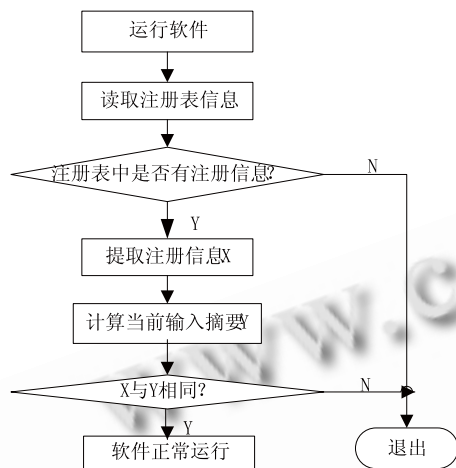


图 3 身份验证流程图

(上接第 177 页)

的集群软件 RAC 进行集群配置。软件的配置布局为，在图 2 的 B1 上安装 1 个数据库和 cluster ware 集群软件，在 B2 和 B3 上分别安装 1 个数据库。具体配置步骤如下：

1) 集群环境准备

• 配置文件修改。包括/etc/hosts、/etc/sysctl.conf 等文件

- 创建用户/用户组和安装目录
- 配置 SSH，修改 oracle 用户配置文件
- 为共享磁盘分区
- 安装配置 OCFS2 和 ASM

2) 安装集群软件

3) 安装数据库软件

4) 创建数据库，并配置网络

5) 集群启动/停止

经过以上三步，一个完整的基于虚拟化的服务器集群实例配置完成。

获取注册信息的关键语句：

首先获得 hkey_current_user\software 子键：

```
key = My.Computer.Registry.CurrentUser.  
OpenSubKey("software", False)
```

```
mykey = key.OpenSubKey("users", False)
```

然后读取注册表中的摘要信息：

```
Pswhash=mykey.GetValue("password").ToString
```

4 结束语

本文分析了软件中敏感信息的安全性问题，综合采用数据加密、身份认证等安全机制，给出了 VS2008 环境中的实现过程。所采用的算法加密强度高，破解难度大，可以有效地保护软件的数据信息。

参考文献

- 1 朱鲁华,陈荣良.数据库加密系统的设计与实现.计算机工程,2002,28(8):61-63.
- 2 李彦秋.面向数据库外层的安全技术研究[硕士学位论文].东营:中国石油大学,2009.
- 3 Daemen J, Rijmen V.谷大武,徐胜波译.高级加密标准(AES)算法—Rijndael 的设计.北京:清华大学出版社,2003.31.
- 4 段钢.加密与解密.北京:电子工业出版社,2008.131-139.

5 结束语

在 Linux 平台之上，利用虚拟化技术，结合负载均衡调度算法以及集群技术，可以大幅提高企业现有服务器的利用率，从而保证企业业务的持续性和数据的完整性，节省企业大量的 IT 成本。鉴于该方案带来的优势，相信 Linux 服务器的虚拟化与集群技术在企业服务器这一广阔的领域里会得到广泛的应用。

参考文献

- 1 王建红.浅析 Linux 虚拟化技术.湖北师范学院学报(自然科学版),2008,28(1):62-64.
- 2 庞辽军,王力,李慧贤.基于集群技术的 Linux 虚拟服务器.计算机工程与应用,2003,39(14):161-163.
- 3 张静.RAID 技术工作原理分析及实现.科技信息(学术版),2008,(32):329-330.
- 4 何禹,胡宇鸿,王一波.虚拟化技术在校园网数据中心的应用.电子科技大学学报,2007,36(6):1461-1464.
- 5 林纲,张治辉.Linux 内核地址映射机制分析及实现.计算机与数字工程,2005,33(7):118-120.
- 6 陈品华.虚拟化技术在中小企业的应用.微计算机信息,2010,3(5):220-221.