基于Linux的高可用集群系统的设计及实现

孟相武[1]，程 劲[2]，罗克露[1]，韩 淙[1]

(1. 电子科技大学计算机学院 成都 610054; 2. 空军成都局驻成飞军代表室 成都 610000)

【摘要】利用服务器集群技术设计了基于Linux操作系统的高可用集群处理系统，描述了系统模型，介绍了系统设计思想，给出了系统实现。对不同的功能部件分离的设计和实现，有效地减少了系统复杂性，降低了错误检测的难度并充分考虑了应用服务器的多样性、复杂性。测试结果表明，系统能够有效保证服务器的可靠运行和企业核心计算的安全。

关 键 词 高可用性; 应用服务器; 集群系统; 节点; 系统可靠性

中图分类号 TP368.5 文献标识码 A

Design and Implement of a High Available Server Cluster System Based on Linux

MENG Xiang-wu，CHENG Jing，LUO Ke-lu，HAN Cong

(1. School of Computer Science and Engineering, UEST of China Chengdu 610054;

2. Military Representative Office(CAC)of Chengdu Bureau, Air Force China Chendu 610054)

Abstract With server cluster technology, a high available cluster system based on Linux is designed. This thesis expounds the system's model、designing idea and implement. In system designing, each part of the system is designed separately to reduce system's design complexity and to make it easier to inspect system's error. Diversity and complexity of application servers are fully considered. Test proves that the system can ensure servers' reliable run and security of enterprise users' kernel computing effectively.

Key words high availability; application server; cluster system; node; system reliability

服务器作为应用部门数据处理的核心部件，存放应用部门中的关键数据。服务器中软、硬件的失效将造成数据的丢失或服务器之间数据不一致，导致巨大的经济损失。保证关键服务器连续、稳定的运行成为计算机应用的迫切需要。服务器集群技术是解决上述问题的一种重要技术。它是指一组共同工作的计算机集合，系统中的任何单台、或多台计算机发生故障均不会导致系统提供的服务不可用[1]。利用集群技术构建的系统具有高可用性、高性价比、高可靠性、扩充性能好等特点。本文设计并实现基于Linux操作系统的高可用性集群系统，以满足企业对服务器的特定工作要求。

1 系统模型

本集群系统由两个节点组成，1) 节点作为工作机，为用户提供特定的服务；2) 节点作为备份，实时监测工作机的工作状态。工作机发生故障后，备份机在软件控制下自动切换到工作状态，保证用户服务不因工作机的故障而中断。

系统中存在控制节点和备份节点两台主机，称为集群主机。控制节点作为集群系统中的工作机，响应用户发出的请求；备份节点监控控制节点的运行状态。当控制节点失效时，备份节点接管控制节点的工作。保证服务的稳定正常运行。控制节点和备份节点的划分是动态的，即，在系统发生异常时，备份节点将转

收稿日期：2003 – 09 – 10
作者简介：孟相武(1972 –)，男，在职硕士生，主要从事网络安全和实时系统方面的研究.

化为控制节点，响应用户发出的请求。系统组成如图1所示。通信网络传输用户请求，以及服务器对用户请求的响应。同步网络采用冗余的方式传输集群系统的控制消息，避免同步网络故障造成集群系统中节点状态不一致；不同的同步网络采用不同的传输接口，避免操作系统的接口软件错误的影响[1]。应用服务器根据用户需求提供具体业务的服务，如：HTTP服务器为用户提供Web服务。

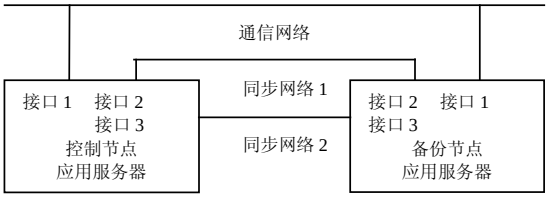

图1 系统组成图

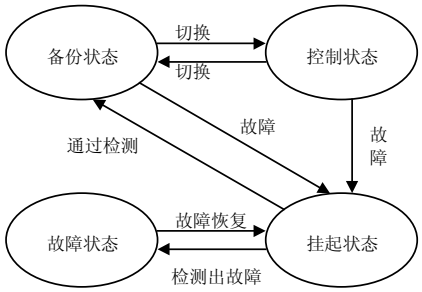

图2 集群处理状态机

2 系统设计

集群是一种低级的系统服务组件[2]，不同的应用服务器构筑在它之上。系统设计必须简洁、高效。简洁性保证系统健壮，利于维护和调试，减轻程序员负担；高效性使集群对系统性能的影响最小化。

2.1 工作模式

系统中存在“工作机－备份机”和“工作机－工作机”两种工作模式。1) 在第一种模式下，所有客户请求都由工作机处理，备份机只检测工作的状态。工作机正常工作时，备份机资源处于空闲状态，优点是实现简单。2) 在第二种模式下，工作机和备份机都同时处理用户请求。其中一台失效时，由另一台单独处理所有用户请求并等待工作机的故障恢复，优点是充分利用了系统资源。但第二种模式必须考虑多台服务器的负载均衡问题，即：两台服务器收到用户请求的多少将影响集群服务器的效率。另一个问题是“有状态”服务器的同步问题。即，服务器上保存了客户访问的状态信息，发生故障切换后，状态信息将丢失，引起服务器的状态不同步。系统采用第二种工作模式，通过开发多种调度算法解决负载均衡问题，并在节点之间设置同步进程同步服务器状态，解决“有状态”服务器的同步问题。

2.2 SSI

在通信网络中，整个集群系统具有一台主机的映像，称为单一映像(Signal System Image，SSI)[1]。单一映像能力保证集群系统对用户透明，用户应用程序不需要任何修改就可以访问集群服务器。例如：在IP网络中，集群系统只存在一个IP地址，称为虚拟IP地址。控制节点发生故障时，备份节点“抢占” 虚拟IP地址。用户业务请求就传送给备份节点(新控制节点)，保证用户应用程序在访问集群系统与访问非集群系统不作任何修改。为了保证系统的SSI，在系统发生切换时，首先，故障主机放弃虚拟IP地址的配置，而新控制节点动态配置虚拟IP地址。然后，利用ARP协议的gratuitous arp消息，改写同一局域网中其他主机ARP协议缓冲存储器，完成IP地址“抢占”。

2.3 有限状态机

系统中每个节点存在四种状态，即：控制状态、备份状态、挂起状态和故障状态。节点之间正常状态消息不影响节点状态，影响节点状态的消息有：切换、故障、通过检测、检测出故障和故障恢复等5条消息。节点状态转换如图2所示。

2.4 通信协议

在高可用集群系统中，节点间周期性地交换状态信息维持集群成员关系，维护整个集群系统状态的一致性。1) 希望故障发生后，系统能在最短时间内将状态消息传送给集群系统中的其他节点，以便快速排除发生故障的节点；2) 希望在正常工作时有较长的周期值，减少系统的开销。

较短的周期值会增加系统通信量。采用单播通信，在每个状态信息交换周期中，通信网络上有n^2个状态信息数据包。采用广播协议，通信网络上只有n个状态信息数据包。为了减少通信量，系统采用广播方式传

输状态信息。

采用广播方式传输集群状态消息有发送者主动和接收者主动两种基本方式。1) 在发送者主动广播协议中，发送者利用TCP/IP协议的广播协议发送数据包到状态消息通信网络，集群系统中所有主机接收广播包，并发送确认包给发送者；2) 发送者维持一个定时器和保存最近发送的数据包，在规定时间内，发送者没有接收到集群系统中某个成员节点的确认包，则发送者认为该节点发生故障。缺点是在正常通信情况下，大量确认消息占用了通信带宽。另外，如果冒充发送者发送广播包，冒充者只需发送一条消息，而发送者必须处理多个应答数据包，很容易受到拒绝服务攻击。

采用接收者主动方式时，接收者接收到错误消息、或接收超时才给发送者发送消息。在正常工作方式下，状态通信网络上只有发送者的状态通知消息，没有接收者的应答消息，减少了状态交换消息的通信开销。系统采用第二种方式传输状态交换消息。

3 系统实现

3.1 系统体系结构

系统根据各个部分的功能进行模块划分，模块组成如图3所示。控制模块是整个系统的核心，负责和系统中其他主机通信，维护系统中各台主机状态的一致性。控制模块周期性地交换状态信息。发生异常时，控制模块调用其他模块调整本节点状态到正常的工作状态。控制模块还要检测本机的运行状态，本机处于不稳定状态时，发送消息给集群系统中其他节点，同时调整本节点状态。

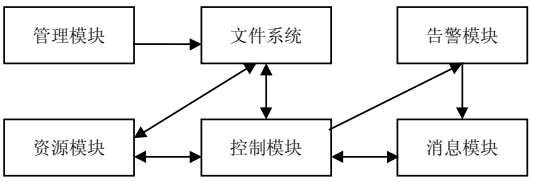

图3　高可用性系统体系结构

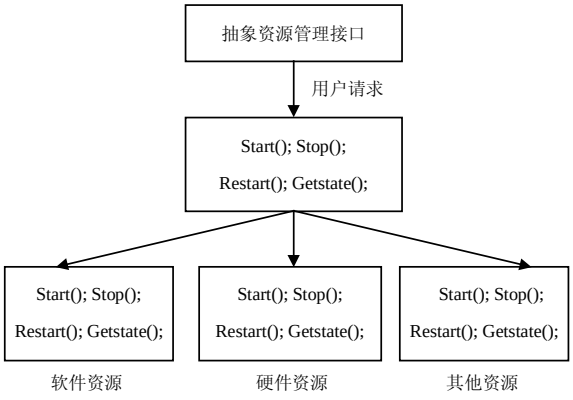

图4　接口管理实现

消息模块实现集群之间、各部件之间的消息交换。进程间用PF_UNIX套接口，状态信息交换用PF_INET的广播通信接口，系统的服务器的状态检测用PF_INET的udp协议接口。消息模块中实现了一个统一的消息交换接口，接口的操作映射到前面的不同实际接口上。

资源管理模块管理整个集群系统中的资源，在发生状态切换后，需要放弃不再使用的资源，或者获得必需的系统资源。而且，需要检测系统中各种管理资源的状态，及时发现系统异常。管理模块保证在系统环境发生变化时，调整系统到最佳状态。系统管理利用配置文件静态配置，利用Linux操作系统的/proc文件系统动态配置[3~5]。告警模块支持email、sms、log、beep等多种告警方法，主要用于立即、快速通知系统的异常状态。

3.2 资源管理实现

3.2.1 接口操作

系统资源包括硬件资源和软件资源，软件资源主要由服务器运行需要的各种支撑程序、系统程序、程序库等构成；硬件资源包括CPU、memory、I/O等服务器必需的运行环境。在集群系统中，必须统一管理这些资源，保证服务器可靠、稳定地运行。然而，各种资源的工作方式差异很大。例如：除了软件和硬件资源，还存在IP地址这类资源。各种资源管理形式和实现方法大不相同，有些软件资源可能由很多服务器共享，

它们对外提供的接口千差万别。这种差异性使得系统实现非常复杂。为此，系统设计了一个抽象资源管理接口，为其他部件提供统一的资源管理接口。资源管理接口操作如图4所示。例如：在发生状态切换的时候，只需要下面的代码：

```
for all resource in system
resource.start();
```

3.2.2 接口数据结构

系统资源管理数据结构如下：

```
struct  resource_ {
struct resource *next;
char name[NAME_MAX];
        int flags;
        int (* start) (void);          启动资源
        int (* stop) (void);           停止资源
        int (* restart) (void);            重新启动资源
int (* get_state) (void);     查询资源的使用情况
}
static struct resource  resources;  /* 系统需要管理的资源 */
```

系统管理的所有资源是可配置的。系统启动时，读取配置文件中的项目注册到系统的资源表中(static struct resource *resources)。这种模式可以适应各种不同服务器对资源的需求，提高系统扩充能力。

4 结 束 语

系统采用了易于扩充的结构，在Linux操作系统上实现了一个两节点高可用集群系统，利用Ethernet的广播功能传输集群控制消息，减少了系统开销。集群中节点状态采用有限自动机设计，通信模块和控制模块采用独立进程实现，增强了系统可靠性。

在测试中，对系统采用手工拔控制节点、备份节点的网线及串口通信线；分别停止控制进程，通信进程；突然关闭节点电源，重启动节点的操作系统等多手段模拟了系统故障。系统均能正确地从故障状态切换到正常工作状态，切换时间不大于10 s。

模拟测试表明系统能够满足企业关键服务器的长期、稳定运行，显著提升服务器的可用时间。测试中也发现了一些问题，例如：对于某些时间敏感的应用，系统故障切换时间相对较长。另外，系统设计没有考虑异构系统的集群处理。这些问题将在下一步工作中予以解决。

参 考 文 献

[1] Knight S, Weaver D, Whipple D, et al. Virtual router redundancy protocol[S]. RFC2338, 1998

[2] Richard M R. Enterprise enabled Linux, Standard RAS implementation[J]. Digital Technical Journal, 2000, 1(4):3 - 15

[3] 李善平, 刘文峰, 李程远, 等. Linux 2.4源代码分析大全[M]. 北京: 机械工业出版社, 2001

[4] Bovet D P, Cesatr M, 著. 深入理解Linux内核[M]第2版. 陈莉君, 冯 锐, 刘欣源, 译. 北京: 中国电力出版社, 2001

[5] Tuexen M, Xie Q, Stewart R, et al. Architecture for reliable server pooling[S]. RFC3237, 2002

编 辑 孙晓丹