

基于 Linux 的电信高可用软件的设计与实现^{*}

左天军¹ 左圆圆² 陈 平¹

(西安电子科技大学软件工程研究所 西安710071)¹ (大唐电信)²

摘 要 许多关键业务应用要求系统能够持续稳定地运行以便为用户提供不间断的服务。高可用系统通过配置冗余硬件和故障监控处理软件,可以大幅度地提高系统的可用性。本文设计实现了一个基于 Motorola CPX8216 和 HA-Linux 的电信高可用应用软件。实验证明该软件能够自动检测系统控制节点和服务进程的错误、失效,并且当发生这种情况时能够自动将系统控制和服务切换到备用节点。文中最后指出了需要进一步研究的问题。

关键词 电信应用,关键业务应用,高可用性,CPX8216,HA-Linux,容错设计

The Design and Implementation of High-Availability Telecom Application Based on Linux

ZUO Tian-Jun¹ ZUO Yuan-Yuan² CHEN Ping¹

(Institute of software Engineering, Xidian University, Xi'an 710071)¹

(Datang Telecommunication)²

Abstract Mission critical applications require systems running stably and persistently to provide continuous services for customers. Combined with redundant hardware and failure handling software, highly available systems can improve system availability greatly. In this paper, we design and implement a high-availability telecom application based on Motorola CPX8216 and HA-Linux. The experiment shows that our implementation can detect errors and failures of system control nodes and service processes. While failures occurring, system control and service can be switched to standby nodes automatically. We give the future research work finally.

Keywords Telecom applications, Mission critical applications, High availability, CPX8216, HA-Linux, Fault-tolerant design

1 引言

目前,在许多应用系统上运行着关键业务,需要不间断地为客户提供服务,即使发生短暂的业务中断,也会导致难以估量的损失。因此,高可用性(High Availability)越来越受到人们的重视。高可用系统的目标是保证系统在没有人工干预的情况下,能够自动地发出告警并解决故障,从而减少突发性数据丢失或服务中断的可能性^[7]。电信网络是一个典型的高可用性系统,它要求系统能够提供99.999%的高可用性,即一年当中业务中断最多只能5分钟^[3]。本文讨论了一种基于 Linux 的高可用系统开发平台,在此平台上设计实现了一个电信高可用应用软件,当系统出现故障时,可以自动将系统控制和服务切换到备用节点,为电信领域中的高端应用提供99.999%的高可用性。

本文第2节概述 Motorola 公司基于 Linux 的高可用系统平台,包括硬件平台、配置方式和软件平台;第3节讨论软件的设计和实现,给出软件体系结构及测试结果;最后总结全文并指出开发应用级高可用软件的改进方向。

2 高可用系统开发平台

本文所研究的高可用应用软件开发硬件平台为 Motorola CPX8216,软件平台为 MCG HA-Linux。

2.1 CPX8216硬件平台

CPX8216是一种16槽位的基于 CPCI 总线结构、并具有高可用性的电信级设备^[5]。系统具有两个独立的6槽位的 CPCI I/O 域,提供了冗余的系统 CPU 板和热交换控制器 HSC。CPX8216标准系统包括两个8槽位的子系统(域),每一个子系统包括2个槽位的系统处理板和6个槽位的非系统 I/O 板,如图1所示。

CPX8216提供了双系统模块、基于 CPCI 的双 I/O 域、冗余的电源和风扇模块,以及冗余的外设。总之,所有的系统组件都有冗余备用,并且支持热交换,使得系统在其部分组件失效时能够继续运行,在不中断应用服务的情况下,对故障组件进行维修。根据用户不同的需求,可以对 CPX8216系统处理板、I/O 板和外设进行多种灵活的配置。下面主要介绍系统处理板的三种配置模式。

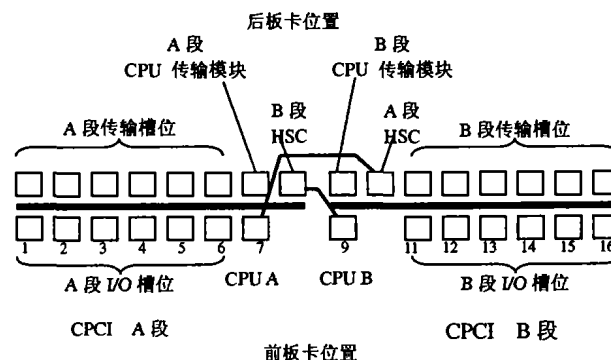


图1 CPX8216标准系统配置

1. 单系统板配置 CPX8216本身具有可灵活配置的特点,因此能够提供不同层次的冗余和可用性。对于不需要完全高可用性的应用程序,可以将 CPX8216配置为单系统板、16槽位的系统。这种配置仅具有冗余的电源以及 CPX8216系统的监控功能。

2. 双系统板主备用配置 在双系统板主备用配置模式下,一个系统板管理全部12个 I/O 槽位,另一个系统板作为热备用,当主用系统发生故障时,备用将接管整个系统。该配置模式如图2所示。

^{*}基金项目:可信软件工程(413150501)。左天军 博士生,主要研究方向软件工程技术,操作系统,网络分布计算;左圆圆 硕士;陈 平 教授,博士生导师。

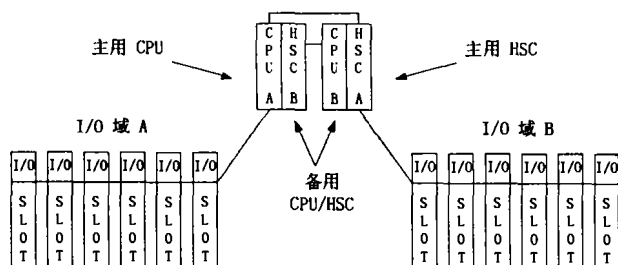


图2 双系统板主备用配置

3. 双系统板负荷分担配置 在双系统板负荷分担配置模式下,每个系统板管理6个I/O槽位,如果其中一个系统板发生故障时,另一个系统板将控制全部的12个I/O槽位。值得注意的是,任何一个系统板都必须随时准备承担另一个系统板的负荷,所以系统关键任务的总和不能超过单个系统板的处理能力。

2.2 软件平台

Motorola Computer Group (MCG)为高可用、嵌入式计算平台设计了HA-Linux^[5],主要用于电信设备中。HA-Linux提供了5个九(99.999%)的可用性,实现了所有系统组件的热交换,包括系统板、I/O板、电源、风扇以及外设。目前基于Linux的需要5个九的服务可用性的应用包括操作维护管理平台、呼叫服务器、媒体服务器、IP网关以及HLR等。在CPX8216硬件平台上,HA-Linux控制并管理系统的热交换组件,保证应用软件在系统部分组件发生故障时依然能够提供正常的服务。

HA-Linux的关键部分在于它的热交换功能,由CPCI-HOTSWAP软件包实现。CPCI-HOTSWAP软件包与Linux内核PCI服务软件包协同工作,完成CPCI系统上Linux的热交换功能。HA-Linux通过系统配置和事件管理软件(SCEM)为用户提供综合的配置、事件、告警和可用性管理等功能。SCEM不仅包括了Linux内核组件,也包括了Linux应用组件。Linux内核组件主要包括PCI内核服务、HSC驱动程序、ENUM设备驱动、EM驱动、和支持HA的设备驱动等。Linux应用组件主要包括可用性管理器、事件管理器、配置管理器和告警管理等。

HA-Linux还提供了系统间通信服务(ISCs)。通过ISCs连接,应用程序能够在两个系统域之间进行数据通信。ISCs为开发HA应用所提供的服务主要有:应用程序间的消息数据传递、远端程序执行、文件传送、心跳检测、检查点、日志以及API和库文件。

3 HA 应用软件的设计与实现

3.1 设计思想

本文在HA-Linux的基础上实现了一个HA应用软件,用于监控CPX8216系统板上的程序运行状态,为I/O板与系统板提供不间断的数据通信服务,并将处理后的数据写入共享磁盘阵列^[10]。

系统的主要设计思想通过配置冗余硬件^[2,8,9]和故障监控处理软件^[6]以大幅度地提高系统的可用性。设计中,系统的硬件采用CPX8216双系统板主备用的配置方式,当主用系统发生故障时,备用将接管整个系统。系统的软件监控模型如图3所示。

在系统板与I/O板之间通过以太网或CPCI背板总线进行通信;在主用系统板与备用系统板之间可以配置以太网和

串口两个通道,一个作为Primary,一个作为Secondary。系统板之间通过HA-Linux提供的ISCs服务传送消息,进行心跳检测。当Primary通道失效时,监控软件可以自动切换到Secondary通道并恢复ISCs连接。

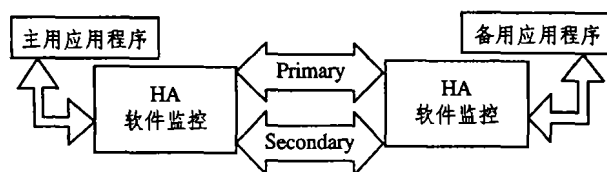


图3 应用软件监控模型

3.2 软件体系结构

运行在系统板上的软件主要包括两部分:应用软件和HA监控软件,其结构如图4所示。应用软件主要实现系统板与I/O板的通信和数据处理。HA监控主要实现应用软件故障监测和系统切换控制。其中,故障监测模块的主要功能为:定时查询本端应用软件模块的运行状态;通过ISCs连接服务发送心跳消息给对端的故障监测模块;当一定时间内没有收到对端的确认消息,则认为对端系统发生故障。在这种情况下,如果本端作为主用系统,则发送备用板故障告警消息给操作维护中心;如果本端作为备用系统,则发送切换命令给本端的HA切换模块进行倒机,在切换完成后,上报操作维护中心。HA切换模块实现系统的切换,并使用ARP欺骗的方法实现IP地址的接管。

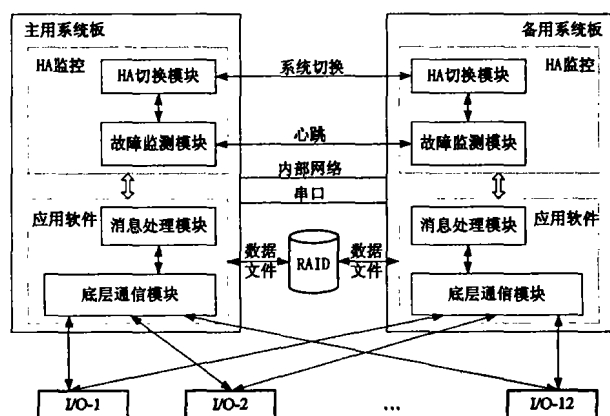


图4 系统板服务监控软件体系结构

因此,通过主用系统板和备用系统板之间的互相监听,可以实现系统的热备份。当CPX8216系统启动时,A系统板和B系统板各自查询本板所控制的I/O域,得到主用或备用的运行状态。假设A系统板作为主用,B系统板作为备用,那么当A系统板出现问题时,B系统板能够自动接替工作并升级为主用系统板,控制整个CPX8216系统,提供系统板与I/O板的数据通信,对I/O板发送到系统板的消息进行处理,并将处理后的数据保存到共享磁盘阵列。

3.3 实现

根据系统板上的软件模块划分,可以相应地实现为两个进程:应用软件进程和HA监控进程,进程之间采用了消息队列进行通信。应用软件进程实现与I/O板的通信和消息处理。HA监控进程是一个守护进程,实时监控应用软件的运行情况,并与对端HA监控进程通过心跳消息互相监测对方的状态。应用软件进程对于不同的应用系统需求有不同的实现方法,HA监控进程对于不同的系统其功能实现非常相似,具有一定的可重用性。

HA 监控进程在系统中占有重要的地位。如果要保证整个系统的高可用性,那么首先必须保证 HA 监控进程的稳定性。在 HA 监控进程的实现过程中,尽可能做到功能简单清楚,减少与其他应用软件的依赖关系,作为一个相对独立的模块运行,这样能够有助于其稳定性的提高。

对程序的测试结果表明,系统从备用切换为主用所需时间小于100ms,在系统切换过程中,I/O 板与系统板之间的通信基本不受影响。在发生故障的系统板恢复正常后,重新查询系统状态,作为备用系统开始运行。

结论 随着计算机和网络的飞速发展,计算机软件在各行各业得到了越来越广泛的研究和应用,尤其在一些关键行业的关键应用上,要求系统必须具有保护业务关键数据和维持应用程序的高可用性的能力。高可用系统对硬件和软件都有一定的要求,在硬件方面,要求系统具有冗余硬件,并且能够在不影响系统运行的前提下,实现部件的热插拔;在软件方面,要求系统具有故障监测和故障处理功能,尽可能减少数据丢失和服务中断,提高系统的可用性。

本文所讨论的 Motorola CPX8216 系统在双系统配置下完全符合高可用系统对硬件的要求,而 HA-Linux 则提供了一个集成的 HA 应用软件开发平台。在此平台上,本文详细讨论了一个 HA 电信应用软件的设计与实现,测试结果证明系统能够自动检测系统控制节点和服务进程的错误、失效,并且当发生这种情况时能够自动将系统控制和服务切换到备用节点,以提供不间断的服务。在不同的应用系统中,其 HA 监控

软件具有很大的相似性,因此本文所给出的 HA 监控软件具有一定的可重用性,进一步的改进方向主要有在主备用系统之间采用检查点(check point)技术^[1,4],定期将主用程序的运行状态信息传送给备用系统。当主用系统宕机时,备用系统能够从最近的检查点处恢复运行。另一方面,故障监控和切换程序的稳定性还需进一步提高。

参考文献

- 1 Preslan K, et al. Scalability and Failure Recovery in a Linux Cluster File System. In: 4th Annual Linux Showcase and Conf. Atlanta, Georgia, Oct. 2000. 169~180
- 2 Malkhi D, Reiter M. An Architecture for Survivable Coordination in Large Distributed Systems. IEEE Transactions on Knowledge and Data Engineering, 2000, 12(2): 187~202
- 3 Gray J, Siewiorek D. High-availability computer systems. IEEE Computer, 1991, 24(9): 39~48
- 4 Burke T. High-Availability Cluster Checklist. Linux Journal, no. 80, Dec. 2000
- 5 <http://www.motorola.com/computer/literature>
- 6 Chen L, Avizienis A. N-Version Programming: A Fault-Tolerance Approach to Reliability of Software Operation. Fault Tolerant Computing, FTCS-8, 1978. 3~9
- 7 Cristian F. Understanding Fault-Tolerant Distributed Systems. Communications of the ACM, 1991, 34(2): 57~78
- 8 Fabre J-C, Perennou T. A Metaobject Architecture for Fault-Tolerant Distributed Systems: The FRIENDS Approach. IEEE Trans. on Computers, 1998, 47(1): 78~95
- 9 Dias D M, Kish W, Mukherjee R, Tewari R. A Scalable and Highly Available Web Server. In: Proc. 41st IEEE Intl. Computer Society Conf. (COMPCOM), Feb. 1996. 85~92
- 10 Gray C, Cheriton D. Leases: An efficient fault tolerant mechanism for distributed file cache consistency. In: Proc. 12th Symp. on Operating Systems Principles, Dec. 1989. 202~210

(上接第16页)

术有三种: CORBA、J2EE、Microsoft .Net。这三者在长期的发展过程中,互相借鉴,因此具有相似的体系结构和实现技术^[5~7],应根据具体应用需求进行选择。

J2EE 平台构筑在 Java 语言之上,因此利用了很多 Java 语言及 Java 虚拟机的优势。EJB 构件模型简单易学,基于 J2EE 开发分布式应用具有简便、快速的特点。J2EE 对 Web 应用提供了良好的支持,同时基于 J2EE 开发的应用显然可运行在不同的硬件平台上。J2EE 平台的强大功能、简单易学,以及多家大公司的成熟产品支持无疑使 J2EE 成为开发基于 Web 的企业应用的首选。但 J2EE 显然无法对除 Java 之外的其他语言提供支持。同时还需要指出的是 J2EE 构件模型的缺陷。EJB 构件仅通过远程接口定义了构件向外界提供服务的接口,却没有对其所需的服务进行刻画和定义,从而导致构件间的依赖关系全部隐含在构件的实现代码中。因此,当需要组装重用构件,或移植部分应用时,必须研究构件代码,抽取出该构件与其他构件间的依赖关系,只有在新的环境中提供具有同样接口、同样功能的依赖构件,才可以完成重用以及移植。显然,这种移植性与重用性不能满足人们的需求。值得一提的是, CORBA 构件模型 CCM^[7] (CORBA Component Model), 在构件接口对提供服务和所需服务都进行了刻画,从而真正实现了基于构件的应用组装。

Microsoft .Net 构筑在 Windows 操作系统之上,因此操作系统提供了很多特殊的支持以及性能上的优化。另外 .Net 引入的对多语言的支持,也是优越于 J2EE 的一个特性。但是 .Net 作为一个产品,专属于一家公司,仅针对一个平台,与 J2EE 作为一个开放式规范,多家公司支持,多种产品实现的现状对比起来,使得人们对基于 .Net 应用的日后维护、升级,以及与其它产品的兼容性都有所担忧^[5,6]。

CORBA 的优势在于它对异构性的屏蔽,强大的集成能力,不象 J2EE 局限于 Java 语言,不象 Microsoft .Net 局限于 Windows 操作系统。CORBA 构件模型 CCM, 具有更加完整

的模型结构。CCM 不仅通过刻画 (Facet) 定义了构件向外界提供的服务接口,允许客户从不同角度使用同一构件的不同功能; CCM 还通过接插口机制 (Receptacle) 和事件机制 (Event Source/Event Sink) 分别定义了构件之间的依赖连接关系^[7], 构件的重用性、移植性明显提高。CORBA 缺点也很明显,就是比较复杂,学习周期较长。虽然有产品支持,但缺乏易于使用的开发工具,从而导致开发基于 CORBA 的应用比较困难而且易于出错。CCM 的出现虽然将极大简化基于 CORBA 的服务器端应用开发,但 CCM 规范仍然比较复杂,目前尚存在很多问题,没有成熟的产品,有待研究和完善。

总结 基于构件的 J2EE 中间件明显简化了分布式应用的开发。构件化思想的引入,改变了开发应用的传统模式,由传统的复杂对象开发,建立工程,编译,链接的过程转变为简单构件开发,应用组装,部署的过程。无论从应用开发的效率,质量,还是灵活性、可用性来说,基于构件的应用开发都比传统基于对象的应用开发具备明显优势,而且能充分满足企业分布应用的服务需求。EJB 的简单易用初步展示了构件的无限魅力,但还存在缺陷,对构件自描述能力的深入研究,对构件模型的继续完善,将对软件的开发、重用产生深远影响,并将促进软件产业向大规模工厂制造模式发展。

参考文献

- 1 Sun Microsystems, Inc. Java(tm) 2 Platform Enterprise Edition Specification, v1. 3. Final Draft - 10/20/00
- 2 Sun Microsystems, Inc. Java(tm) 2 Platform Enterprise Edition Specification, v1. 4. Proposed Final Draft 2- 11/8/02
- 3 Roman E. Mastering Enterprise Java Bean. John Wiley & Sons, Inc. 1999
- 4 Roman E. Mastering Enterprise Java Bean. Second Edition. John Wiley & Sons, Inc. 2002
- 5 Roman E, Oberg R. The Technical Benefits of EJB and J2EE technologies over COM+ and Windows DNA. Dec. 1999
- 6 Vawter C, Roman E. J2EE vs. Microsoft .NET: A comparison of building XML-based web services. June, 2001
- 7 Object Management Group. Corba Component Model Specification V3. 0. Formal/2002-06-65