摘 要

随着我国海上交通、船舶运输等行业的迅速发展，海事局 VTS 系统（Vessel Traffic Services）得到了广泛的应用，从而也对 VTS 系统服务器的可靠性提出了越来越高的要求，使之成为了研究热点。本文的重点就是通过对 VTS 双机热备系统建模分析，提出了改进系统检测率的相关方法，实现了具有较高可靠性的 VTS 系统服务器平台。

本文以海事局 VTS 项目为背景，首先在研究双机热备系统相关理论和关键技术的基础上，对 VTS 双机热备系统进行了分析设计。本文采用主从式的工作模式，根据 VTS 系统的特点将每台服务器的结构层次分为操作系统层、双机管理层、应用服务层；并设计和阐述了 VTS 双机热备系统的工作流程及其双机软件的模块功能。其次用基于 Markov 链的马尔柯夫预测法对双机热备系统的可靠性进行研究。在分析和预测双机热备系统工作状态的基础上，建立了相应的系统数学模型，并对模型的微分方程求解，通过 MATLAB 分析可靠度数据和曲线，来说明相关参数对系统可靠性的影响，然后分析了影响 VTS 双机热备系统可靠性的因素，通过加入监控应用进程状态的功能和增加一条心跳链路来改善双机热备系统的检测率。最后针对 VTS 系统，阐述了双机热备系统的具体部署和实现，给出了相关配置文件和脚本，并采用故障注入法对系统进行功能测试。通过改进前后可靠度的比较，验证了系统可靠性的提高。

本文研究实现的双机热备系统已经成功的在秦皇岛、重庆等海事局的 VTS 系统中得到应用。目前，该双机热备系统性能稳定，用户反应良好，保障了 VTS 系统持续运行的同时，满足了其服务器的可靠性要求。

关键词：VTS；双机热备；可靠性；Markov 模型

ABSTRACT

As the rapid development of the China's industry, such as maritime transport, ship transport, the Maritime Bureau VTS (Vessel Traffic Service) system has been widely used. Thus requirements of the reliability of the VTS server also proposed increasing and the reliability of server systems has gradually become a research hotspot. The focus of this paper is to propose to improve the detection rate of the system and implement the server platform of the VTS system by research on the reliability of the dual-machine hot standby system.

The research is based on the VTS project of MSA. Firstly, the paper proposed to achieve the overall design based on the dual-machine hot standby system theory and key technologies. The work mode of hot standby system is master-slave, in accordance with the characteristics of VTS system which consists of three layers of the operating system layer, the dual-machine manage layer and the application service layer. And the design of the working process and the function analysis of dual-machine software modules are described. Secondly, using the Markov forecasting method based on Markov chain, research on the reliability of the VTS dual-machine hot standby system. To establish the system of the corresponding mathematical model based on analysis and prediction in the system state of the work, and solve the differential equations, through the MATLAB analysis of reliability data and curves, to illustrate the relevant parameters on the impact of system reliability , and then analyzed the impact on the reliability of the dual-machine hot standby system factors, by acceding to the process of monitoring the application of state functions and to add a heartbeat link to improve the detection rate of the dual-machine system. Thus enhance the reliability. Finally, in accordance with the VTS system, deploy and achieve the dual-machine hot standby system, and give the specific description of the setting profile and related scripts. Fault injection method is used to carry out functional tests on the system. By reliability comparison before and after the improvement, verify the increase of the reliability.

In this paper, the dual-machine hot standby system has been successful applied in the VTS system of the QinHuangDao, ChongQing and other Maritime Bureau. At

present, the system performance and stability, the user response is good, the protection of the VTS system at the same time continuing to run to meet the reliability requirements of the server.

Key Words: VTS; Dual-Machine Hot Standby; Reliability; Markov Model

大连海事大学学位论文原创性声明和使用授权说明
原创性声明

本人郑重声明：本论文是在导师的指导下,独立进行研究工作所取得的成果，撰写成博/硕士学位论文 “VTS 双机热备系统的可靠性研究与应用” 。除论文中已经注明引用的内容外，对论文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本论文中不包含任何未加明确注明的其他个人或集体已经公开发表或未公开发表的成果。本声明的法律责任由本人承担。

学位论文作者签名：邢佐星

学位论文版权使用授权书

本学位论文作者及指导教师完全了解大连海事大学有关保留、使用研究生学位论文的规定，即：大连海事大学有权保留并向国家有关部门或机构送交学位论文的复印件和电子版，允许论文被查阅和借阅。本人授权大连海事大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，也可采用影印、缩印或扫描等复制手段保存和汇编学位论文。同意将本学位论文收录到《中国优秀博硕士学位论文全文数据库》（中国学术期刊（光盘版）电子杂志社）、《中国学位论文全文数据库》（中国科学技术信息研究所）等数据库中，并以电子出版物形式出版发行和提供信息服务。保密的论文在解密后遵守此规定。

本学位论文属于：　保　密□ 在____年解密后适用本授权书。

不保密☑ （请在以上方框内打“√”）

论文作者签名：邢佐星 导师签名：

日期：09年7月1日

第 1 章 绪 论

1.1 研究背景

1.1.1 VTS 项目简介

在进入二十一世纪以来，海事系统开始逐步实施水上安全监督信息系统工程，海事系统主要包括雷达系统、信息传输系统、船舶交通管理信息系统、显示系统、船岸通信系统等；另外的配套或辅助设施主要包括电源系统、气象系统、闭路电视系统等[1]，如图 1.1。VTS 是 Vessel Traffic Services 的缩写，意为船舶交通服务。VTS 系统一般也叫船舶交通管理信息系统，至今还没有明确的定义。本文讨论的 VTS 系统就是海事系统中的船舶交通管理信息系统。

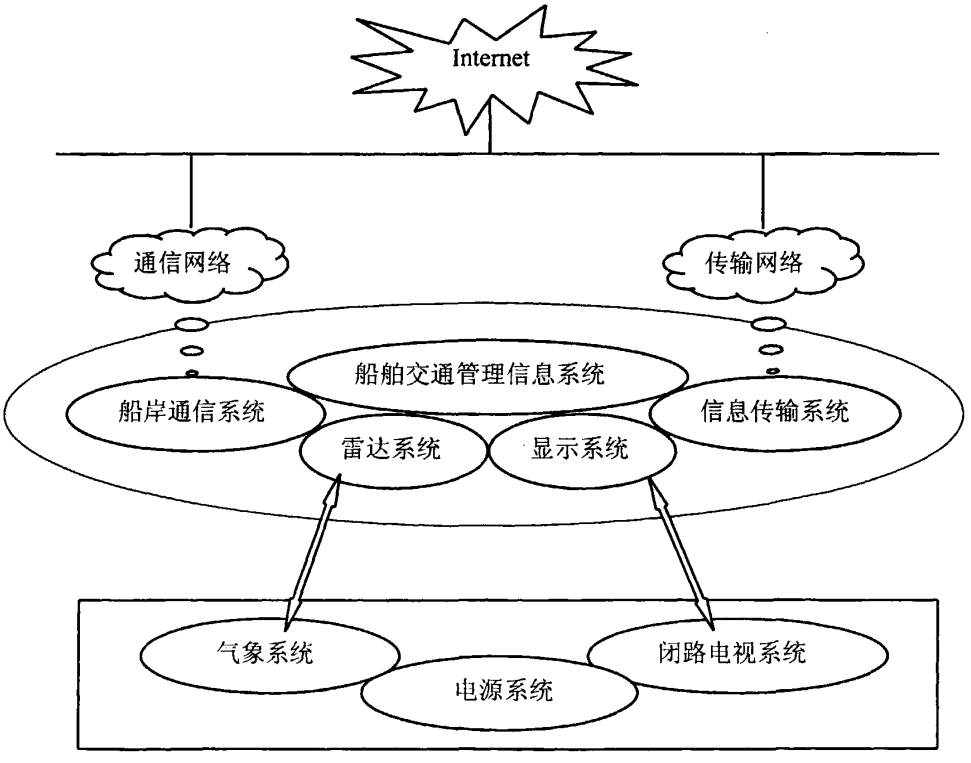

图 1.1 海事系统结构

Fig.1.1 Structure of the maritime system

本文的 VTS 系统是采用 JSP+Servlet 开发的基于 MVC 设计模式 B/S 结构的 WEB 应用程序，并结合当今流行的 Struts，Hibernate 等开源框架和 Ext 控件，具有较强的可移植性，便于维护等优点。主要的功能有：①申报、审批进出港口的船舶。②管理相关船舶的基本信息，并对其它子系统提供数据支持。③解析和显示港口内船舶的动态情况以及天气情况。并且这个系统联合了高科技的海上监视网络，可同时跟踪监视动态和静态目标各 200 个左右。当船舶进入 VTS 区域时，雷达系统自动捕捉目标，其它系统立即计算出其运动参数，VTS 监督员可对船舶进行动态跟踪监视，掌握船舶航行态势，并可通过 VHF 无线电话与船舶交流、沟通，实施交通管理和组织。能为海上交通事故的调查、取证提供宝贵的第一手资料。

1.1.2 VTS 系统服务器情况

随着我国海上交通、船舶运输等行业的快速发展，也给海事局 VTS 系统服务器的性能带来巨大的挑战，由于海事局一些部门的 VTS 系统采用的服务器已经对目前的高性能计算机系统应用服务提供不了相应保障，经常会因为软件或者硬件方面的原因导致服务器故障，进而引起数据丢失等方面问题。然而还有一些部门采用的服务器平台是多点集群的方式，虽然可以对应用服务提供保障，性能也更佳。但是缺乏对操作业务、数据量等方面的考虑，造成计算机资源及部门资金的浪费。

另外，服务器系统需要长时间无故障运行，可靠性缺乏相应的保障，这样会造成应用系统服务的暂停，从而导致业务操作中断等一系列的问题。

1.2 研究目的

1、构建一个既能满足海事局相关部门的业务需求，又能保证 VTS 系统的稳定运行，同时也大大降低运行和维护系统的资金费用的服务器平台，以满足海事局用户的迫切要求。

2、可靠性是衡量服务器系统质量的重要技术指标，在保证 VTS 系统中的应用程序能够长时间的无故障运行的前提下，还应当使服务器系统具有较强的故障检

测以及快速修复的能力，确保服务器系统满足用户提出的高可靠性要求。双机热备系统可以扩展到多点集群，双机热备的可靠性研究成果对集群系统来说也有着很大的借鉴和参考价值。

1.3 研究现状

1.3.1 双机热备现状

目前国内外在服务器平台的构建方面，日益被用户接受和广泛使用的高可用系统是多节点集群，但对于一些中小企业用户来将，服务器的价格相对来讲比较昂贵，多点集群系统所需要的硬件及软件成本很高，加上集群系统技术较双机热备系统的复杂度有所提高，需要更专业的技术人员进行管理，这无形中就增加了系统的维护费用。新用户在选择高可用解决方案时往往已经拥有不止一个关键应用，或者就算某些用户目前只有极少的关键应用，但其考虑未来关键业务数量的增加，仍然可以先购买两个节点的“集群”，也就是通常所说的的双机热备系统，日后可以进行扩展，如何简单的扩充也是用户急需解决的问题，“集群”的“平滑扩展”就能很好的满足用户这些需求。

不久前，日本的 F5 公司开发出了高可用性集群 BIG-IP，它是使用于本地网络站点或数据中心的高可用的、智能化的负载平衡产品，并且提供了对网络流量的自动和智能的管理[2][3]。与其它的高可用集群系统不同的是，BIG-IP 向用户提供的是一个即插即用设备，而其它的提供的都是软件方法。在国内方面，联想公司推出了用于高性能计算分布式NS10000高可用集群服务器，主要基于联想万全4500R服务器，以总体成本相对较低的设备组合，足以替代传统 RISC 小型机和中型机的工作，而价格仅为市场上同等性能小型机的 1/2—1/4[4]。

双机热备或多点集群大多是通过集群软件（对于双机热备来说也叫双机软件）来实现的。那么，目前市场上的高可用集群软件有那些呢？据了解，由于集群的技术含金量比较高，因此能够拥有集群核心开发技术及产品的企业在国际上也较少，而且往往是一些技术实力较强的公司才能推动及支持集群产品的研发。集群软件基本分为三个派系[5]：

1、欧美系列：以 Symantec、EMC 为主，其产品功能较好，产品支持平台较全，但对应用环境要求较高，操作、配置都比较繁琐复杂，产品价格偏高，售后服务成本也相对较高；

2、国内系列：以联鼎软件 LanderCluster 为主，联鼎软件是国内高可用领域历史悠久的著名开发企业，其高可用产品 LanderCluster 的用户众多，在国内各个重要行业都拥有大量成功案例。产品支持平台全面，包括 Windows、Linux、Unix，功能也非常全面。

3、日本系列：以 NEC 的产品为主，支持 Windows 及 Linux 平台，由于 NEC 的产业链较多，集群只是其中很小一部分，因此技术及投入力度相对有限。图 1.2 是初步市场调查得出的各个集群软件派系所占的市场份额。

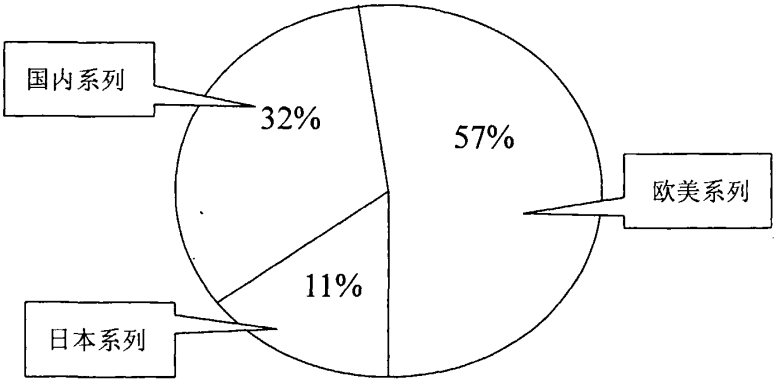

图 1.2 集群软件市场份额

Fig. 1.2 The market share of cluster software

1.3.2 可靠性的研究现状

对计算机系统性能单方面的关注容易使人忽略其他的一些重要的方面，例如可靠性就是一个常常被忽略的因素[6]，由于系统的可靠性较差，往往给人们带来巨大损失，尤其对于关键业务，停机通常是灾难性的，因此停机带来的损失也是巨大的。如表 1.1 所统计的数据，列举了不同类型企业应用系统停机所带来的损失。

表 1.1 各种计算机应用系统宕机损失统计

Tab. 1.1 Loss of computer system downtime statistics

损失（美元/每分钟）	应用系统
43000	呼叫中心（CC）
36000	电子商务（EC）
28000	企业资源计划（ERP）
23000	客户服务系统（CSC）
19000	供应链管理（SCM）

由此可见，服务器系统的可靠性已是非常紧迫且急待解决的问题，在对系统可靠性方面的研究中，建模是一种比较常用而有效的方式。通过对系统进行简化和抽象后，应用现在多种的模型分析方法，可以使我们对系统的整体性能和行为方式有更加具体的分析和预测。在理论和技术的不断发展中，产生了组合模型[7]、动态故障树[8]、神经网络[9]和 Markov 模型[10][11]等多种分析方法。通过分析和比较，并将这种研究成果应用于关键应用的服务器系统上，不同程度地满足和优化了某些应用系统的需求，从而使用户更容易接受价格低廉、应用广泛、性能可靠的服务器系统。

1.4 论文组织结构

论文总共有六个部分。第一部分，在介绍课题背景和研究目的后，阐述了研究现状。第二部分，探讨双机热备的作用和实现方式基础上，深入研究双机热备的关键技术。第三部分，以海事局 VTS 项目为背景，提出了系统实现的设计方案，包括系统设计原则、系统的结构层次设计和工作流程设计等。第四部分，利用马尔柯夫预测法对 VTS 双机热备系统建立了相应的可靠度数学模型，分析相关参数对系统的可靠性影响，并通过改进系统的检测率来提高 VTS 双机热备系统的可靠性。第五部分，给出了 VTS 双机热备系统的具体部署和实现过程，并采用故障注入法对系统进行测试，最后通过比较改进前后可靠度，验证了系统可靠性的提高。第六部分，对全文工作进行总结，指出进一步的工作。

第 2 章 双机热备的相关理论与技术

2.1 双机热备概述

2.1.1 HA 简介

高可用性 HA(High Availability)：指的是通过尽量缩短因日常维护操作（计划）和突发的系统崩溃（非计划）所导致的停机时间，以提高系统或者应用的可用性。它与不间断操作的容错技术有所不同。

高可用性系统是目前企业防止核心计算机系统因故障停机的最有效手段。通过硬件冗余或软件的方法都可以很大程度上提高系统的可用性，硬件冗余主要是通过在系统中维护多个冗余部件如硬盘、网线等来保证工作部件失效时可以继续使用冗余部件来提供服务；而软件的方法是通过软件对系统中多台机器的运行状态进行监测，在某台机器失效时启动备用机器接管失效机器的工作来继续提供服务，所以集群是 HA 系统的主要表现方式。

集群：是由两台或多台节点机（服务器）构成的一种松散耦合的计算节点集合，为用户提供网络服务或应用程序（包括数据库、Web 服务和文件服务等）的单一客户视图，同时提供接近容错机的故障恢复能力。集群系统一般通过两台或多台节点服务器系统通过相应的硬件及软件互连，每个群集节点都是运行其自己进程的独立服务器。这些进程可以彼此通信，对网络客户机来说就像是形成了一个单一系统，协同起来向用户提供应用程序、系统资源和数据。除了作为单一系统提供服务，集群系统还具有恢复服务器级故障的能力。集群系统还可通过在集群中继续增加服务器的方式，从内部增加服务器的处理能力，并通过系统级的冗余提供固有的可靠性。

双机热备系统属于集群的一种，这一概念包括了广义与狭义两种意义[12][13]：

从广义上讲，双机热备系统就是对于重要的服务，使用两台服务器，互相备份，共同执行同一服务。当一台服务器出现故障时，可以由另一台服务器承担服务任务，从而在不需要人工干预的情况下，自动保证服务器系统能持续的提供服务。

从狭义上讲，双机热备系统特指基于 Active/Standby 方式的服务器热备。服务器数据包括数据库数据同时往两台服务器上写，或者使用一个共享的存储设备。在同一时间内只有一台服务器运行即 Active 机器，当其中运行着的一台服务器出现故障无法启动时，就会通过双机软件的侦测（一般是通过心跳检测）将 Standby 机器激活，保证应用在短时间内完全恢复正常使用。

2.1.2 RAID

RAID：RAID 是“Redundant Array of Independent Disk”的缩写，中文意思是独立冗余磁盘阵列（最初为廉价磁盘冗余阵列）冗余磁盘阵列技术诞生于 1987 年，由美国加州大学伯克利分校提出。最初研制目的是为了组合小的廉价磁盘来代替大的昂贵磁盘，以降低大批量数据存储的费用，同时也希望采用冗余信息的方式，使得磁盘失效时不会使对数据的访问受损失，从而开发出一定水平的数据保护技术，并且能适当的提升数据传输速度[14]。

RAID 的优点[15]：①扩大了存贮能力，可由多个硬盘组成容量巨大的存贮空间。②降低了单位容量的成本，市场上最大容量的硬盘每兆容量的价格要大大高于普及型硬盘，因此采用多个普及型硬盘组成的阵列其单位价格要低得多。③提高了存贮速度，单个硬盘速度的提高均受到各个时期的技术条件限制，要更进一步往往是很困难的，而使用 RAID，则可以让多个硬盘同时分摊数据的读或写操作，因此整体速度有成倍地提高。另外，可靠性 RAID 系统可以使用两组硬盘同步完成镜像存贮，这种安全措施对于网络服务器来说是最重要不过的了；容错性 RAID 控制器的一个关键功能就是容错处理，容错阵列中如有单块硬盘出错，不会影响到整体的继续使用，高级 RAID 控制器还具有拯救功能。

RAID 技术规范：RAID 技术主要包含 RAID 0～RAID 7 等数个规范，它们的侧重点各不相同，这里重点介绍下 RAID5。RAID 5 不单独指定奇偶校验磁盘，而是在所有磁盘上交叉地存取数据及奇偶校验信息。方法是将校验数据以循环的方式放在每一个磁盘中；磁盘阵列的第一个磁盘分段是校验值，第二个磁盘至后一个磁盘再折回第一个磁盘的分段是数据，然后第二个磁盘的分段是校验值，从第三个磁盘再折回第二个磁盘的分段是数据，以此类推，直到放完为止。这种方式

能大幅增加小档案的存取性能，不但可同时读取，甚至有可能同时执行多个写入的动作，就是说读/写指针可同时对阵列设备进行操作，提供了更高的数据流量。RAID 5 更适合于小数据块和随机读写的数据。其应用最好是联机应用处理系统，至于用于图像处理等，未必会有最佳的性能[16]。其各种规范的比较如表 2.1。

表 2.1 常见 RAID 技术比较

Tab.2.1 Comparison of common RAID technology

	RAID 0	RAID 1	RAID 3	RAID 5
名称	无差错控制	镜像结构	专用校验条带	校验条带分散
允许故障	否	是	是	是
冗余类型	无	副本	校验	校验
热备用操作	不可	可以	可以	可以
硬盘数量	一个以上	两个	三个以上	三个以上
可用容量	最大	最小	中间	中间
减少容量	无	50%	一个磁盘	一个磁盘
读性能	高	中间	高	高
随机写性能	最高	中间	最低	低
连续写性能	最高	中间	低	最低

另外，我们可以结合多种 RAID 规范来构筑所需的 RAID 阵列，例如 RAID 10 (RAID1+0)、RAID30、RAID50 就是一种应用较为广泛的阵列形式。用户一般可以通过灵活配置磁盘阵列来获得更加符合其要求的磁盘存储系统，比较如表 2.2。

表 2.2 组合 RAID 技术比较

Tab.2.2 Comparison of combination of RAID technology

	RAID 10	RAID 30	RAID 50
名称	跨越镜像阵列	跨越专用校验阵列	跨越分散校验阵列
允许故障	是	是	是
冗余类型	副本	校验	校验
热备用操作	可以	可以	可以
磁盘数量	跨越 2 个阵列	跨越 3 个阵列	跨越 4 个阵列
可用容量	最小	中间	中间
减少容量	50%	一个磁盘	一个磁盘
读性能	中间	高	高
随机写性能	中间	最低	低
连续写性能	低	中间	最低

2.2 作用及实现方式

1、双机热备的作用

目前，国内企业使用的双机热备系统大多是基于中小型计算机系统的，实践证明，双机热备是提高计算机服务器系统可靠性的有力措施。随着国内计算机信息系统的推广和普及，许多中小企业提出了利用 PC 服务器作为计算机主机系统的需求[17]。对于较重要或很重要的应用需求来说，建立双机热备，可以更好的保证服务器系统的安全运行，而双机热备究竟能为我们带来些什么呢？

(1) 提高稳定性

服务器是一种高稳定性的计算机，作为网络的节点存储、处理网络中的数据、信息，它被称为应用服务的灵魂。虽然服务器最大的特点就是它的稳定性超过了一般的台式机，但服务器要想做到 100%不死机或不出问题是不可能的。通过对服务器的双机热备，可大大减少因服务器瘫痪带来的网络瘫痪。因此双机热备技术大大提高了服务器以及网络的稳定性。

(2) 安全保障

对于服务器来说，最需要重视的就是数据安全和服务安全。服务器常见的数据安全保障方法有数据备份及 RAID 等。虽然这些数据备份方案能解决硬盘的数据及服务安全问题，但仍解决不了服务器故障引发的数据安全问题[18]。在采用双机热备后，当一台服务器出现软、硬件故障时，另一台服务器可以在短时间内将故障服务器的职权接管过来，能很快地恢复服务器的应用，保证网络应用服务的持续性。

2、双机热备的实现方式

双机热备有两种实现方式，一种是基于共享存储设备的方式，另一种是没有共享存储设备的方式，一般称为纯软件方式。

(1) 共享存储设备方式

基于共享存储设备的双机热备是双机热备的最标准方案[19][20]。对于这种方式，采用两台服务器，使用共享的存储设备（磁盘阵列柜或存储区域网 SAN）。两台服务器可以采用互备、主从、并行等不同的方式。在工作过程中，两台服务器将

以一个虚拟的 IP 地址对外提供服务，依工作方式的不同，将服务请求发送给其中一台服务器承担。同时，服务器通过心跳线（目前往往采用建立私有网络的方式）侦测另一台服务器的工作状况。当一台服务器出现故障时，另一台服务器根据心跳侦测的情况做出判断，并进行切换，接管服务。对于用户而言，这一过程是全自动的，在很短时间内完成，从而对业务不会造成影响。由于使用共享的存储设备，因此两台服务器使用的实际上是一样的数据，由双机软件对其进行管理。

这种实现方式的主要优点有：

①由于磁盘阵列柜能加快系统 I/O 速度，所以对于 I/O 要求较高的系统运行效率高。

②双机通过共享数据来达到高可用目的，风险集中到磁盘阵列柜上面。而磁盘阵列柜是由很多便宜、容量较小、稳定性较高、速度较慢磁盘，组合成一个大型的磁盘组，利用个别磁盘提供数据所产生的加成效果来提升整个磁盘系统的效能。前面已经介绍过其相关知识和优点，这里不再赘述。

其基本结构如图 2.1 所示。

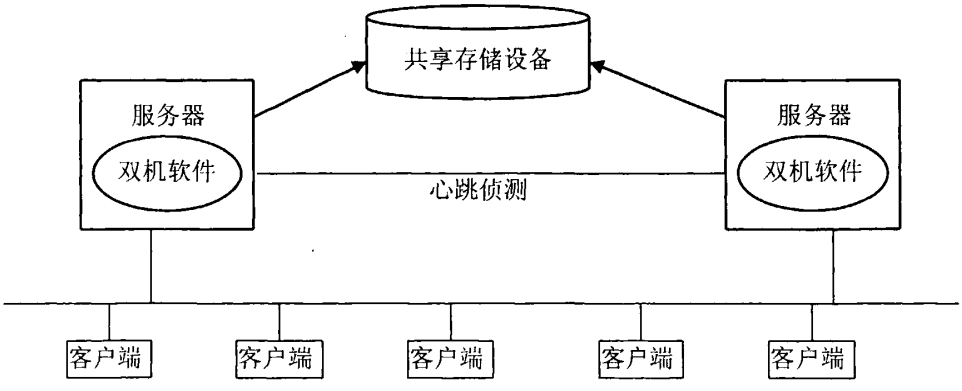

图 2.1 共享存储设备的双机热备

Fig. 2.1 Shared storage device of the dual-machine hot standby

(2) 纯软件方式

对于纯软件的方式[21]，则是通过镜像软件，将数据可以实时复制到另一台服务器上，这样同样的数据就在两台服务器上各存在一份，如果一台服务器出现故障，可以及时切换到另一台服务器。这一方式不受距离的限制，但会产生数据的前后不一致或数据库读取的速度会受一定的影响，如图 2.2。纯软件方式还有另外一种情况，即服务器只是提供应用服务，而并不保存数据（比如只进行某些计算，做为应用服务器使用）。这种情况下同样也不需要使用共享的存储设备，而可以直接使用双机或集群软件即可。但这种情况其实与镜像软件无关，只不过是标准的双机热备一种小的变化。

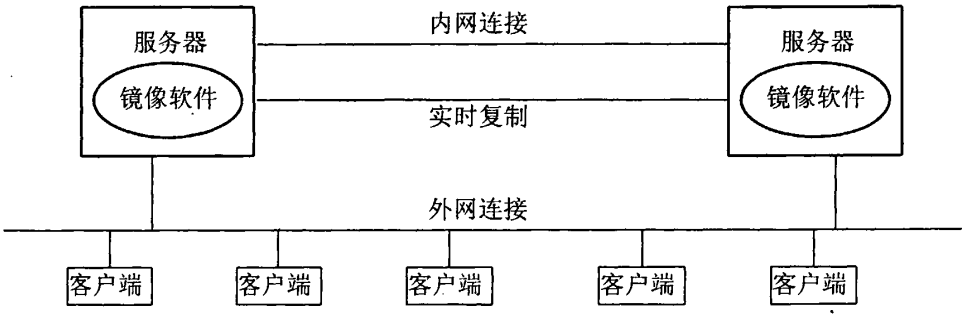

图 2.2 纯软件方式的双机热备

Fig. 2.2 Pure software of the dual-machine hot standby

纯软件方式的优点：

①避免了磁盘阵列的单点故障，对于双机热备，本身即是防范由于单个设备的故障导致服务中断，但磁盘阵列恰恰又形成了一个新的单点。（比如，服务器的可靠系数是 99.9%, 磁盘阵列的可靠系数是 99.95%，则纯软双机的可靠系数是 99.9%*99.9%=99.99%，而基于磁盘阵列的双机热备系统的可靠系数则会是略低于 99.95%）。

②节约投资，不需购买昂贵的磁盘阵列。

③不受距离的限制，两台服务器不需受磁盘电缆的长度限制（光纤通道的磁盘阵列也不受距离限制，但投资会大得多）。这样，可以更灵活地部署服务器，包括通过物理位置的距离来提高安全性。

但纯软件方式有非常明显的缺点：

①可靠性相对较差，两台服务器之间的数据实时复制是一个比较脆弱的环节。

②一旦某台服务器出现中断，恢复后还要进行比较复杂的数据同步恢复。并且，这个时段系统处于无保护状态。

③没有事务机制，由于其复制是在文件和磁盘层进行的，复制是否成功不会影响数据库事务操作，因此有出现数据不完整变化的情况，这个存在着相当的风险。

因此，建议除非不得已，不要选择纯软件方案。何况现在市面上大多采用共享存储的方式来实现双机热备系统，纯软件方式以前应用得较少，主要一方面是由于当时市场上比较流行的双机软件不支持纯软件方式，另一方面是由于少数支持纯软件方式的产品其可靠性不太令人放心。

所以在进行双机热备时，如果投资充裕、数据量大（1T 以上），可以采用共享磁盘阵列柜的方式，并且应尽量选择高可靠性（如著名品牌的）设备。当然，本文由于具有一定的实验条件，以往也没有采用纯软件的方式实现过双机热备系统的经验和对软件稳定性的信心，故本文不采用纯软件的方式，而采用共享存储设备的方式来实现双机热备系统。

2.3 双机热备关键技术

在双机热备系统中，常用的关键技术有故障诊断、检查点机制、证实策略、任务接管等。

2.3.1 故障诊断

故障诊断机制[22]中最常用的模式是报告式、问答式以及根据它们的形式作出各种变化的模式。

1、报告式

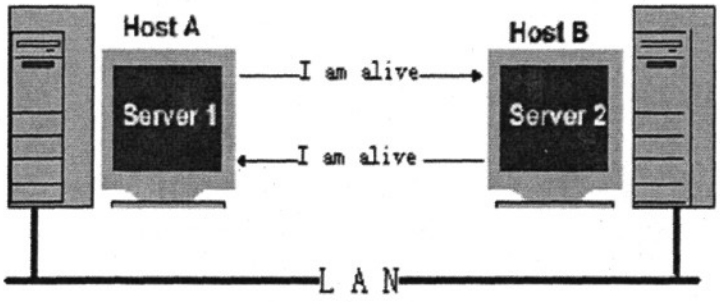

图 2.3 报告式故障诊断

Fig. 2.3 Report fault detection

在报告式中，如图2.3。被监测节点是活动的，它会周期性的发送心跳(Heartbeat)数据信息以通知监测节点它仍然处于正常状态。如果监测节点在一定限制时间内没有收到被监测节点的心跳信息，它则怀疑该节点已失效。因为信息数据的传送在系统内只有一个方向的（对于某一台服务器节点来说），所以它的效率较高。可以用硬件的多播(multicast)机制来实现多个监测节点同时监测相同对象，被监测节点周期性的发送心跳数据信息给监测节点，只要收到信息，监测节点设置一个时间限制，如果在接收到同一被监测节点发来的心跳信息之前超过了这个时间限制，则触发失效事件。

2、问答式

在问答式中，信息数据的方向与控制数据的方向是相反的。如图 2.4 所示，在这种模式中，被监控节点是被动的。监测节点周期性的发送“Are you alive?”请求给被监测节点。如果被监测节点回应“I am alive”，则表明其仍然处于良好状态。因为对被监测节点而言有两个方向的信息数据发送，所以这种模式可能比报告式效率低点，但对于应用开发者来说，因为被监测节点是被动的，而且不需要任何关于时间的知识，这种模式较为方便使用，例如它们并不需要知道监测节点希望收到数据的频率，即报告式中的心跳周期。

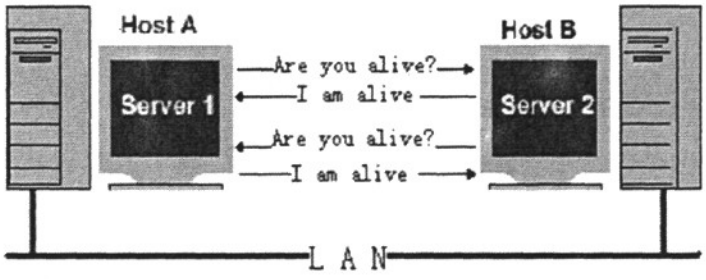

图 2.4 问答式故障诊断

Fig. 2.4 Ask and answer fault detection

3、混合式

混合式组合了以上两种模式，在这种模式里面，报告式和问答式可以在同类对象上同时使用。诊断过程分成两个不同的阶段，在第一阶段中，所有被监测节点假设使用报告式，因此发送心跳数据。在一段延迟后，监测转为第二阶段，在这个阶段里，假设所有在第一阶段中没有发送心跳数据的被监测节点使用问答式，监测节点发送 aliveness 信息给每个被监测节点，并且期望从被监测节点上收到心跳数据，如图 2.5 所示。

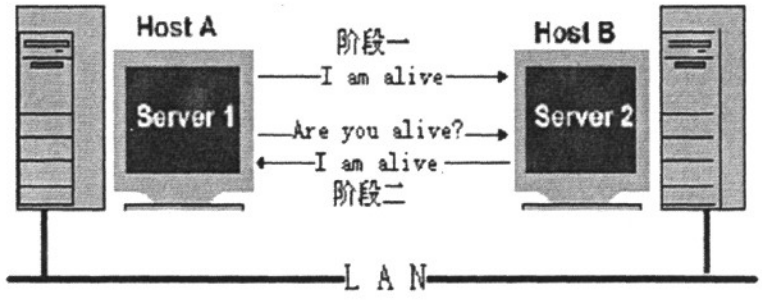

图 2.5 混合式故障诊断

Fig. 2.5 Mixed fault detection

如果被监测节点没有在一定时间限制内发送这种信息，则假定其已失效。混合式在本质上并不是一种新的故障诊断模式。它可以被看成是一种混合不同监控类型的方法，即它不需要监测节点知道每个被监测节点支持哪种诊断模式。因此它提供了更多的灵活性，让被监测节点使用最合适的信息交互模式。

以上就是最常见的三种故障诊断模式，它们有着各自的特色，并在不同的应用中有着不同的效率。根据网络拓扑结构和应用的通讯方式，选择不同的模式对系统的性能有着很大的影响。

2.3.2 检查点机制

检查点机制[23][24]是指为了能够在程序执行到中间出故障后不必从头开始，周期性地设置检查点以保存中间状态，一旦发生故障，可以从最近的检查点重新执行，这种检查点设置与卷回方法是容错中常采用的一种软件技术。系统发生故障后，将相关进程回滚到故障前系统一致性状态（检查点），经过状态恢复后从该检查点处重新执行（而不是从程序开始执行），实现对系统故障的恢复；从而节省了大量重复计算时间。这种基于检查点的后向恢复技术不仅可以对系统瞬时故障进行自动恢复，也是恢复未知故障（在某一应用设计过程中未预料到的故障）的唯一手段，如图 2.6 所示。

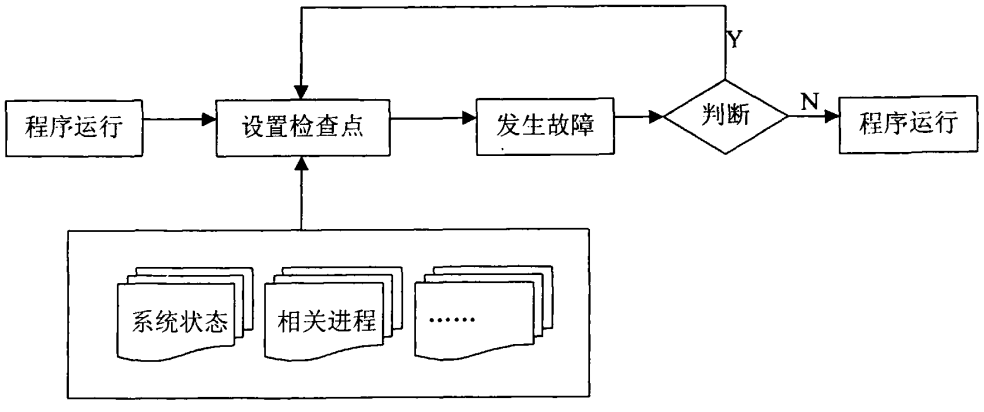

图 2.6 检查点机制

Fig. 2.6 Checkpoint mechanism

当错误发生时，使用检查点可使受影响的进程从最后一次保存的检查点（状态）而不是进程开始重新运行。这个技术特别适合于保护长时间运行程序中出现短暂错误的情况。长时间运行的应用程序通常是运行数天或数周的处理数字程序，对于这种应用程序重新开始而言，即便恢复，但由错误造成的偶然损坏也是不可接受的。

检查点可以是透明的并且在运行时自动插入，或者由应用程序的程序员手工插入。在透明的方法中，检查点由处理器地址状态的全局快照组成，包括操作系统的所有动态数据。其他透明的方法包括处理器内部描述表、栈以及静态和动态数据段。另一方面，在手工方法中，程序员负责精确定义哪些数据对应用程序确实是关键性的，这样可以显著的减少检查点的规模。

2.3.3 证实策略

要使双机热备系统的备份节点应用程序按照主节点的执行轨迹运行。不能单纯依赖于消息到达的实际顺序，因为每个处理节点都同时从多个端口接收消息，虽然采用通讯协议可以保证主备节点从同一个端口接收到的消息顺序一致，但端口之间消息的顺序就无法保证。为此，可以采用证实的策略来保证主备节点应用执行轨迹的一致性，如图 2.7。

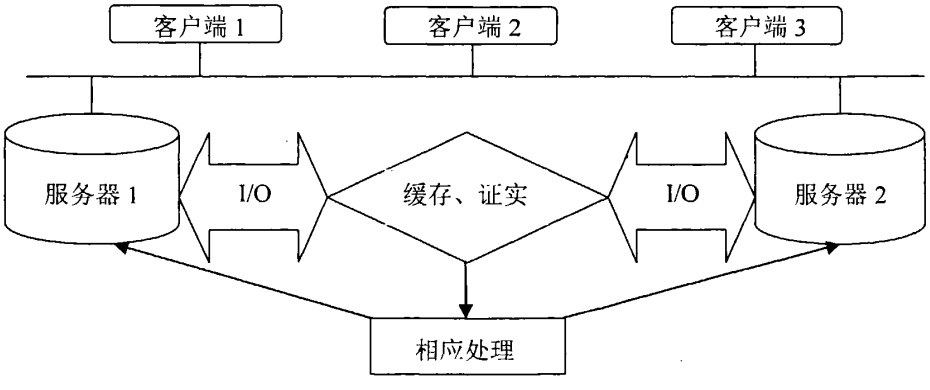

图 2.7 证实策略

Fig. 2.7 The strategy to confirm

当主节点运行到一个同步点时，向备份节点发送一条证实消息，表明刚才处理的是哪一条消息，备份节点对接收到的原始消息进行缓存而不送给应用程序处理。只有收到主节点的证实消息后才将得到匹配的原始消息送给相应的应用程序进行处理。

2.3.4 任务接管

任务接管(failover)[25]是双机热备系统恢复功能的核心。这里先说明下系统出现错误后的恢复技术，错误恢复技术主要有前向恢复和后向恢复两种。前向恢复技术指的是系统从故障中恢复时，从出错时刻以后的某一时刻点开始恢复。后向恢复技术指的是系统从故障恢复时，退回到以前的某一个状态，重新开始处理。采用后向恢复方案中，系统的周期性为运行在双机热备系统的进程中保存检查点信息，发生故障后系统回滚到故障发生处，如图 2.8 所示。在独立于应用程序的可移植方式下后向恢复较容易实现，并已被广泛采用。然而回滚的时间开销问题是一个应仔细考虑的问题；同时，后向恢复需要避免出现多米诺效应。

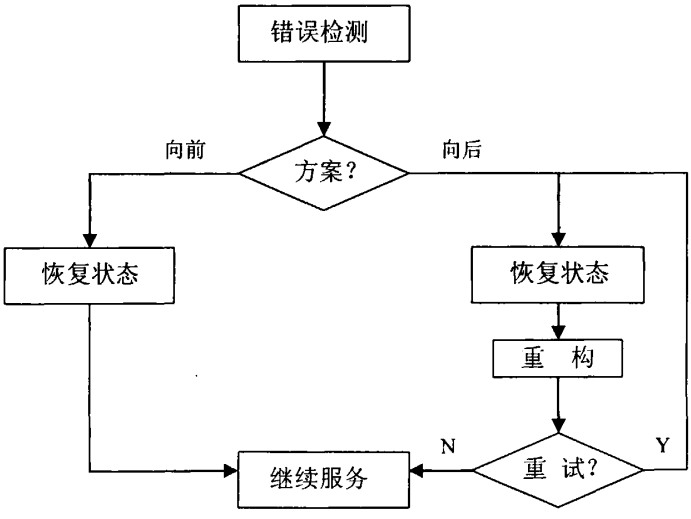

图 2.8 错误恢复技术

Fig.2.8 Techniques of fault recovery

如果执行时间是一个很重要的参数，比如在实时性系统中不能容忍回滚恢复花掉如此长的执行时间，此时应采用前向恢复方案。这个方案中，系统不是回滚到故障前的某个检查点；相反，系统利用故障诊断信息构建一个有效的系统状态，继续执行下去。前向恢复依赖于应用程序且可能需要额外的硬件设备加以支持。

任务接管应该是完全透明的，不需要管理员的干预或用户手动重新连接。任务接管也能有效地用于另一个目的：维护操作，它可以简单的通过将一个服务器上保护的应用服务切换到第二个服务器上来实现，这样就实现了系统的在线维护，并且减少甚至消除了普通维护任务的检修时间以及操作系统或其他应用软件升级所带来的服务器停机。现在这一点相当的重要，因为系统不可用的最大单方面的因素就是维护或升级造成的。

第 3 章 VTS 双机热备系统的分析设计

在前一章中介绍了双机热备系统的相关理论和关键技术。本文采用共享存储设备的方式即通过在两台服务器上运行双机软件和共享磁盘阵列来对 VTS 双机热备系统进行分析设计，其的物理架构如图 3.1。

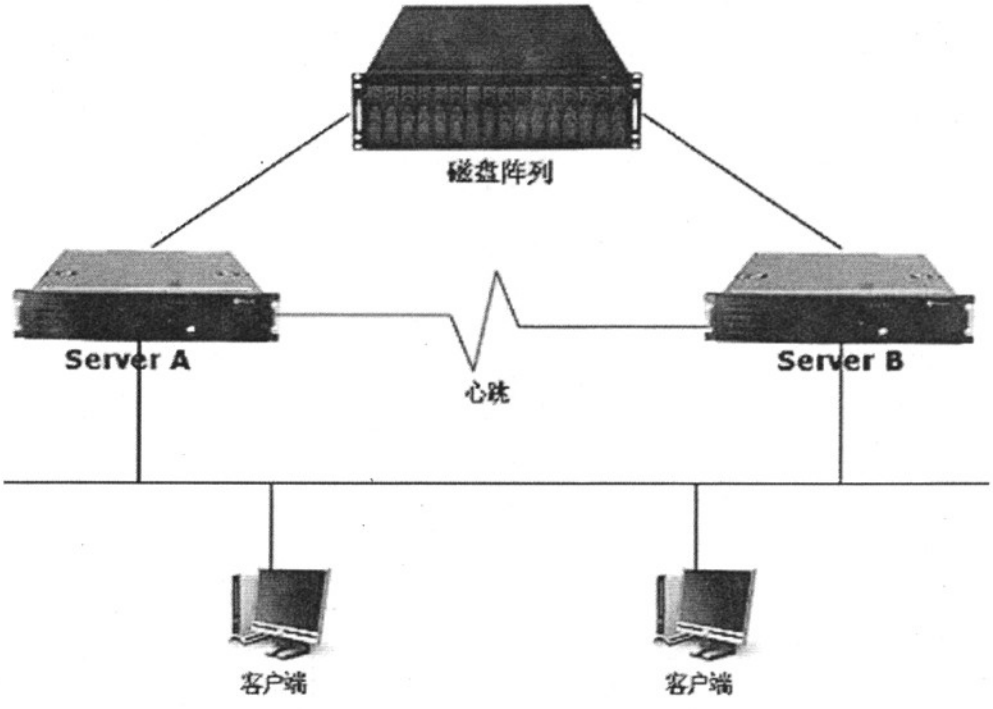

图 3.1 VTS 双机热备物理架构图

Fig. 3.1 Physical structure of the VTS dual-machine hot standby system

3.1 VTS 双机热备系统的设计原则

本文根据海事局 VTS 系统的要求在设计双机热备系统时主要遵循以下几个方面的原则。

（1）硬、软件设计模块化，扩展、重组方便灵活；

（2）仲裁切换逻辑单元智能化，以双机软件实现自动判决；

（3）避免不必要的无效切换，即切换执行后能够保证系统正常工作；

（4）主机能够侦测到备份机故障，并发出相关通知；

（5）要求确保切换单元正常工作并且具有可靠的自测试、自检测功能。

3.2 VTS 双机热备系统的工作模式

双机热备系统常见的工作模式有三种：主从式双机热备模式、双机互备模式、双机双工模式。

主从式双机热备份模式是一台服务器为工作机(ActiveServer)，另一台服务器为备份机(StandbyServer)，在系统正常情况下，工作机为应用系统提供支持，备份机监视工作机的运行情况（工作机也同时监视备份机是否正常，有时备份机因某种原因出现异常，工作机应尽早通知系统管理员解决，确保下一次切换的可靠性），如图 3.2 所示。当工作机出现异常，不能支持应用系统运行时，备份机主动接管(TakeOver)工作机的全部工作[26]，继续支持应用系统的运行，从而保证提供不间断的应用服务。

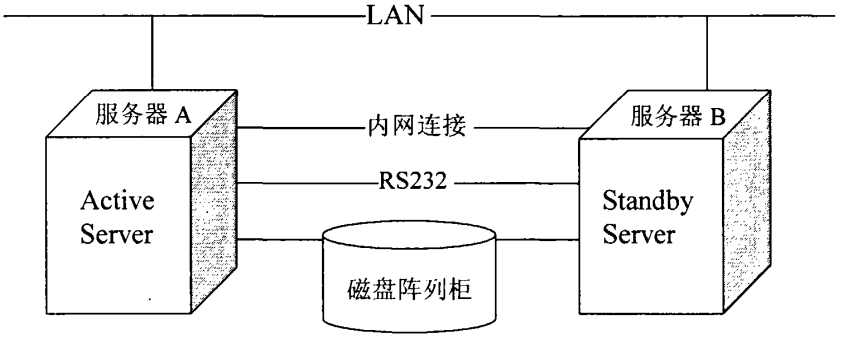

图 3.2 主从式双机热备示意图

Fig. 3.2 The master-slave dual-machine hot standby system

另外还有两种模式是双机互备和双机双工。所谓双机互备就是两台服务器均为工作机，在正常情况下，两台工作机均为应用系统提供支持，并互相监视对方的运行情况，参见图 3.3。当某一台工作机出现异常，不能支持应用系统正常运行

时，另一台工作机则主动接管异常机的应用，从而保证应用系统能够不间断的运行。但是，当一台工作机出现异常并被接管后，正常运行的工作机负载会随之加大，严重的情况下有可能影响到应用系统的响应速度。所以此时必须尽快修复异常机，以缩短双机系统单机运行的时间。

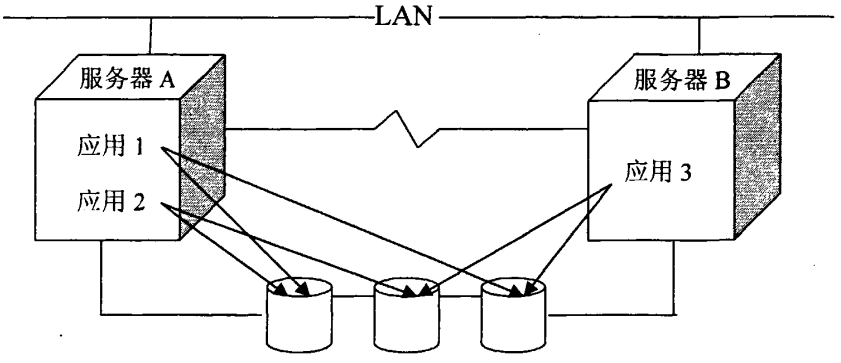

图 3.3 双机互备示意图

Fig. 3.3 The dual-machine each prepared system

双机双工模式也是两台服务器均为工作机(ActiveServer)，在系统正常的情况下，它们并行同步响应外部服务请求，并对服务同步进行处理。在处理过程中，分不同阶段对处理的中间结果进行对比，同时给出每个中间结果的评估。在最后输出的时候根据所有评估和当前的结果表决策略得出唯一的输出结果。可以看出，当一台工作机出现异常时，不能支持应用系统正常运行，整个双机双工模式失效，输出结果不进行表决，因此也应该尽量避免单机运行的情况.

由此可见，虽然主从式双机热备模式并没有充分利用资源，但就应用系统的稳定运行而言，与双机互备和双机双工模式相比，具有较高的可靠性。

由于海事局可以提供充足的硬件资源，而且对应用系统有严格高可靠性要求。不仅要保证主机系统能够 24 小时提供不间断的服务，还要求发生故障切换时，应用系统的性能和响应速度不受影响，以确保网络系统、网络服务、共享磁盘空间、共享文件系统、进程以及数据库的高速持续运转。所以本文采用的工作模式是主从式双机热备模式。

3.3 VTS 双机热备系统的层次结构设计

根据 VTS 船舶交通管理信息系统的特点，本文的双机热备系统分为操作系统、双机管理、应用服务三个层次结构，各层次结构如图 3.4。

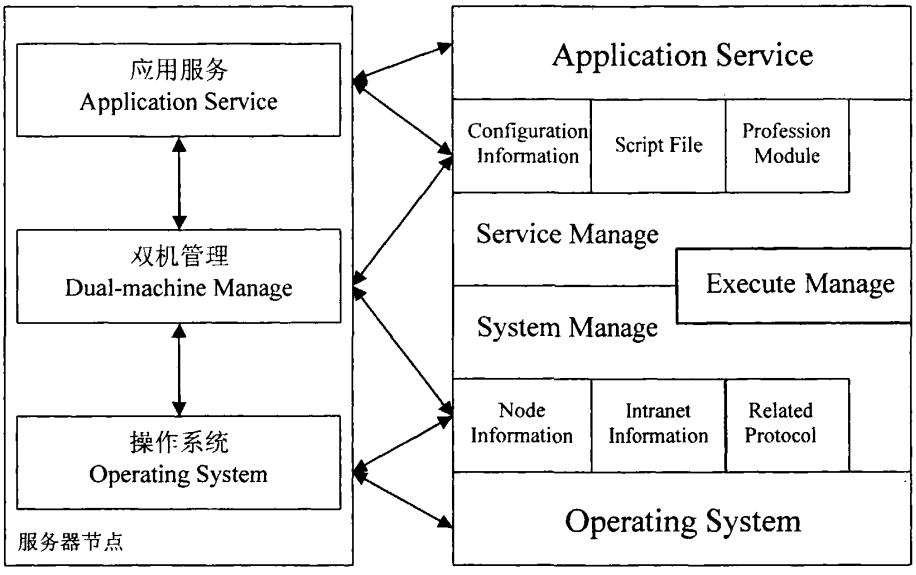

图 3.4 VTS 双机热备系统层次结构

Fig. 3.4 Structural level of the VTS dual-machine hot standby system

3.3.1 操作系统层

操作系统层。这是构建 VTS 双机热备系统所必须的基础，在系统中的两个服务器节点的操作系统要求版本一致（本文使用的是 Red Hat Linux AS 4）。它提供 VTS 双机热备系统所必须的网络环境和文件系统等，根据应用服务的不同，提供支持 VTS 系统所必须的硬件设备及相关的存储环境（如存储子系统上划分的文件系统等）。

3.3.2 双机管理层

双机管理层。在该管理层中，分为三个部分，System Manage、Service Manage

以及 Execute Manage。

系统管理(System Manage)的主要功能有：①节点管理；根据 VTS 双机热备系统当前的环境信息，如服务器节点的硬件信息(Node Information)，对 VTS 双机热备系统进行控制，通过对节点硬件信息监测，来判断整个 VTS 双机热备系统状态是正常状态，还是非正常状态[27]，是否能够正常对外提供服务。②网络管理；对双机热备系统的网络状态信息(Intranet Information)进行侦测，根据当前系统提供的网络信息，针对网络的故障进行系统状态处理。③系统管理；通过相关协议(Related Protocol)通知其它的处理方式进行相应的处理。

服务管理(Service Manage)的主要功能有：①配置管理；对应用系统即 VTS 系统的配置文件进行控制，通过配置文件(Configuration Information)将应用系统部署到双机热备系统上，并保障应用系统的稳定运行。②脚本管理；因为操作系统层中采用的是 Linux 操作系统，可以通过脚本文件(Script File)来启动或者停止相关应用服务，还可以利用脚本来完成双机热备系统某些特定的要求。③模块管理(Profession Module)；对于双机软件来说可以提供相应 API 即二次开发的接口，来优化加强 VTS 双机热备系统的性能。

执行管理(Execute Manage)的主要功能有：①连接系统管理和服务管理并对其提供的相关数据信息等执行后续操作，使 VTS 双机热备系统无论是对操作系统服务还是应用系统服务都会作出相应的反应。②对 VTS 双机热备系统的故障，进行判断和执行相关处理，并且对切换后系统的状态进行定位组合，提供相关的数据信息使得系统管理和服务管理继续发挥各自的作用。

3.3.3 应用服务层

应用服务层。在该层中主要针对 VTS 系统的功能即所提供的应用服务进行管理和控制。双机管理层中的服务管理根据配置文件对该层中的应用服务进行定位，通过脚本文件来启动和停止该层中的应用服务。当服务的相关程序启动完成以后，应用服务层根据对 VTS 系统的定义，对定位好的应用系统提供服务，当其中的应用进程因故障丢失时，通知双机管理层对任务进行移交，保障 VTS 双机热备系统中的应用服务稳定运行。

3.4 VTS 双机热备系统的工作流程设计

3.4.1 启动流程

系统启动时，对于操作系统、应用对象资源的启动次序，是有逻辑依赖关系的，包括 VTS 系统应用服务的启动顺序。因此，资源启动的步骤，会参照双机管理层的功能来执行，如图 3.5。VTS 双机热备系统先启动服务器节点的 CPU、内存、硬盘等资源，然后启动应用系统的服务资源等等。

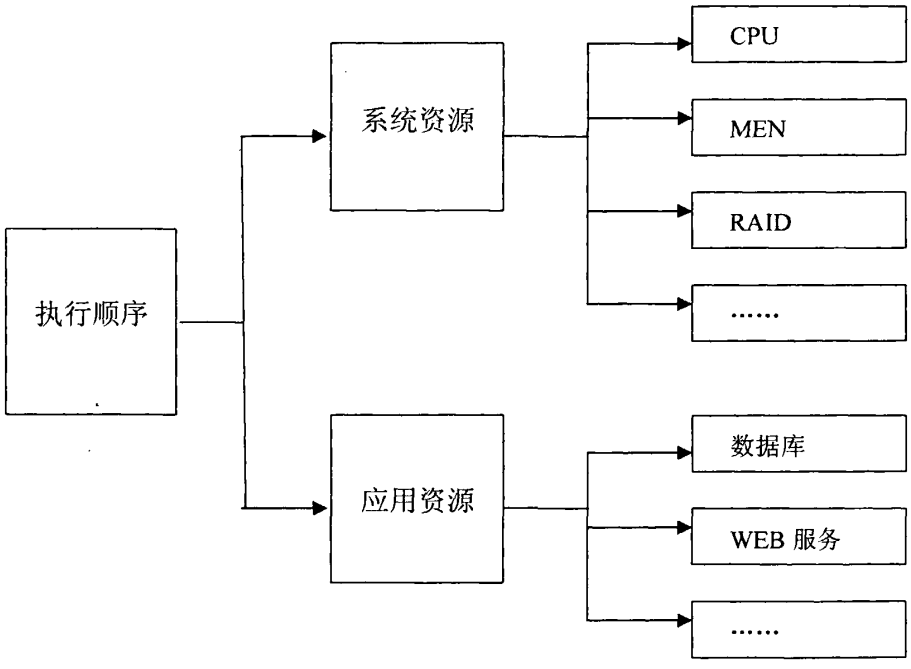

图 3.5 启动流程

Fig. 3.5 Start process

3.4.2 加载流程

系统在启动后，首先读取 VTS 双机热备系统的配置文件，该文件是系统在启动后由双机管理层中的功能模块生成，它完成的是一个类似于服务器节点注册的功能，该配置文件描述的是各自服务器节点的相关信息等参数，其数据结构如下：

struct nodeinfo

```
{
    char nodename[MAXNAME];   //服务器节点名称
    char nodeip[IPADD];          //服务器节点 IP
    int cpufre;     //CPU 主频
    int mem;        //内存大小
    int cache;      //缓存大小
    ……
}nodeinfo;
```

然后根据系统的配置信息，进行双机热备系统的状态组合。根据当前的网络状态和系统参数，对服务器节点进行调整，建立双机热备系统的初始状态[28]。如图 3.6。

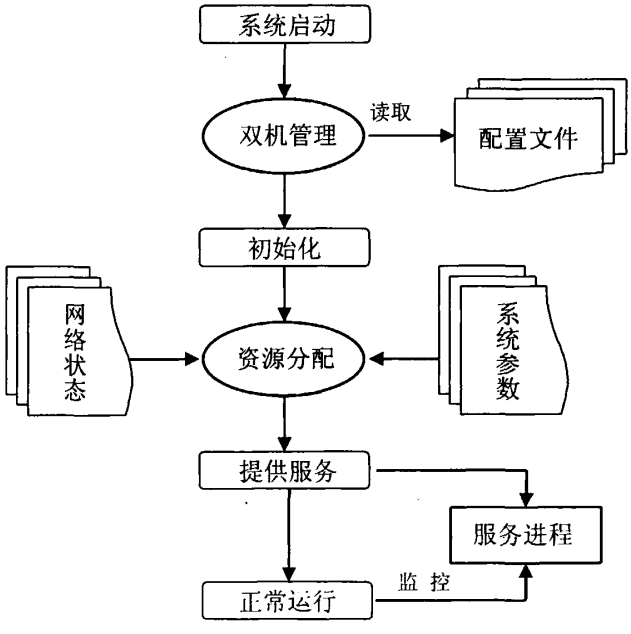

图 3.6 加载流程

Fig. 3.6 Load process

当初始状态建立起来后，双机管理层中的系统管理向执行管理提交各服务器节点的状态信息，执行管理根据服务器节点的网络状态和系统中对资源的定义，

对双机热备系统中的节点进行资源分配，使双机热备系统中的某个服务器节点获得对外提供应用服务的资源。

系统应用服务任务启动后，在双机管理层中启动监控功能，对所启动任务的关键进程进行监控，保障对外提供服务资源的健康。当以上资源建立起来后，双机热备系统进入正常运行状态。

3.4.3 心跳流程

在进入正常运行状态后，通过专用的通讯链路即心跳检测，采用的是内网 TCP/IP 和串口 RS232 两条心跳链路，来和双机热备系统中的另外一个服务器节点进行通讯，传输服务器节点的状态信息，使双机热备系统中每个服务器节点获得均获得本节点和对方节点的实时状态。

当系统中服务器节点故障时，双机管理层中的执行管理根据当前的状态和该故障节点在双机热备系统中的角色做出判断，如果该服务器节点为工作机时，双机热备系统会自动将属于该服务器节点的资源和任务移交到备份服务器节点，保证系统资源和应用服务正常运行；如果该服务器节点为备份机，则需要通知双机热备系统对备份服务器节点进行调整维护，将该故障节点从双机热备系统表中删除。这时整个系统属于单机运行状态，直到该备份服务器节点修复后重新纳入到系统表中，恢复其备份机的角色。

双机热备系统运行后，在双机管理层中会创建控制进程、状态进程、所有心跳传输介质的读进程和写进程[29]，心跳数据的传输主要是通过这几个进程来完成，传输过程如下：

(1) 状态进程启动后，会创建一个时钟。状态进程根据时钟信号，周期性地把本服务器节点状态信息打成数据包，写入 FIFO。心跳数据包的数据结构：

```
struct heartbeatdata
{
    char nodename[MAXNAME];   //服务器节点名称
    char nodeip[IPADD];          //服务器节点 IP
    struct Inode inode;      //服务器节点网络信息数据
```

```
struct Snode snode;     //服务器节点硬件信息数据
struct Minfo minfo;     //任务信息数据
}heartbeatdata;
```

(2) 控制进程从 FIFO 中读出数据包，同时写入状态管道和通讯管道。状态管道主要包括本服务器节点和对方节点状态信息的数据包，通讯管道用于心跳传输介质写进程和控制进程之间的通讯，每种心跳传输介质分别有一个通讯管道。

(3) 心跳传输介质写进程从通讯管道中读出数据包，通过心跳传输介质发送给另一个节点。

(4) 心跳传输介质读进程从传输介质读取另一个节点发送来的数据包，写入状态管道。如图 3.7。

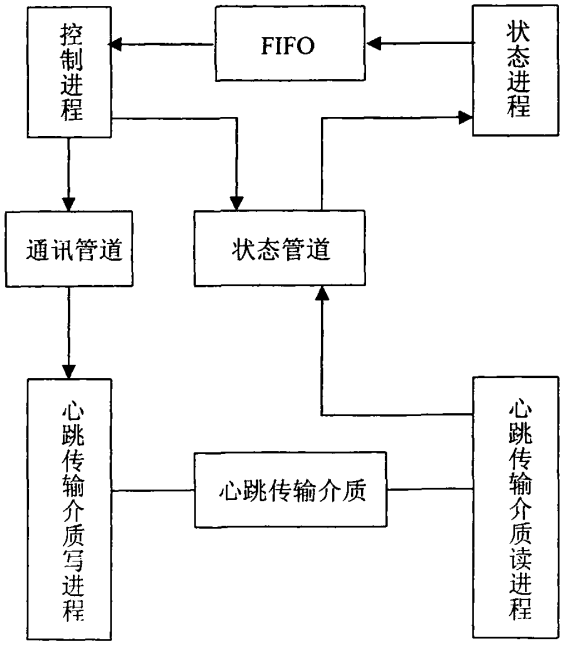

图 3.7 心跳数据流

Fig. 3.7 Heartbeat data stream

这样，状态进程可以从状态管道获得两个服务器节点的状态信息，并通过其

它渠道进行相关的判断和处理。状态进程使用本地时间记录获得节点状态信息的时间，每当状态进程获得服务器节点状态信息，状态进程使用当前时间更新相应节点的时间记录 T。状态进程利用时间记录判定节点是否失效，使用公式：$T_0=T_1-T_2$，计算出可以判定服务器节点失效的时间值 T_0，其中 T_1 表示当前时间，T_2 表示可以认定节点失效的时间间隔。如果 T 不小于 T_0，认为节点正常，否则认定服务器节点已经失效。

也就是说，如果服务器节点在限定的时间内没有接收到本服务器节点或另一个服务器节点发送来的状态信息，则判定本节点或另一个节点失效。另外，状态进程可以对状态信息的数据包进行解析，来判断本节点或对方节点的状态是否正常，如有问题则判定相应的服务器节点失效。

3.4.4 切换流程

当主服务器节点故障时，备份服务器节点在设定的时间间隔内没有收到主服务器的“I am alive”即心跳数据包，等待一定时间 T_2 后如果仍然收不到则认为主服务器节点失效；则进行服务器节点切换，自动接管主服务器的 IP 地址和应用服务等资源[30]，继续提供应用系统服务。

接管 IP 地址是指双机热备系统心跳流程开始时，主服务器节点会建立一个浮动 IP 地址的接口，用来被外部用户访问。当这个节点失效时，另一个节点会开始一个同样 IP 地址的接口，并且使用地址解析协议来确保所有通过这个地址的通信都被这台服务器节点接收。主服务器节点停止的应用服务和备份服务器节点启动的应用服务通过执行相应脚本 stop.sh 和 start.sh 来完成。

脚本的基本内容格式如下：

```
#bash
su - user -c "application program or someting" //执行某些应用程序
sleep n     //休眠几秒
su – otheruser -c "application program" //切换到另一个用户下，执行应用程序
……..
```

在备份服务器节点代替故障主服务器节点工作后，故障的服务器可离线进行

修复工作。在故障修复后，主服务器节点将以备份服务器节点的身份重新进入双机热备系统，而原来的备份服务器将作为主服务器节点继续提供应用服务（即主备角色互换），恢复成故障前的 VTS 双机热备系统正常工作时的状态。切换过程如图 3.8 所示。

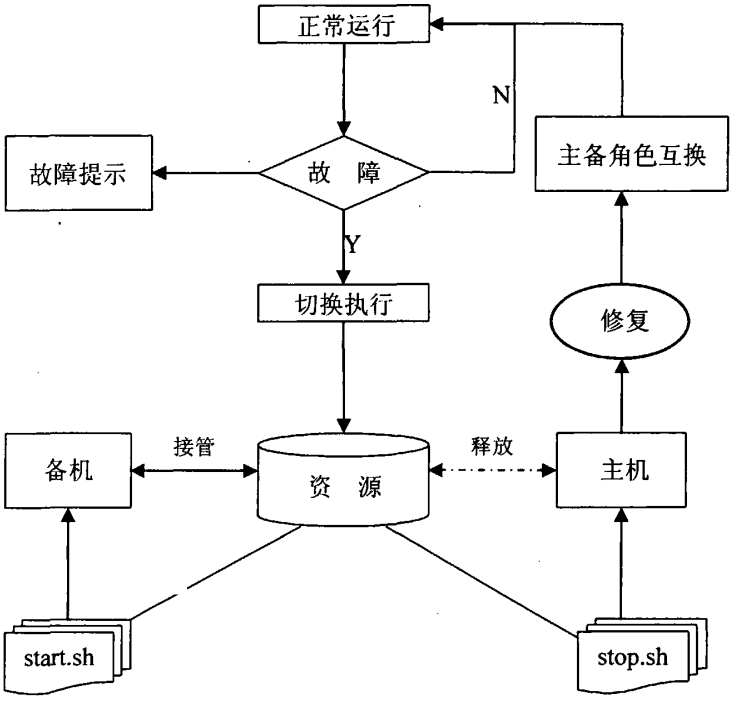

图 3.8 切换流程

Fig. 3.8 Switch process

3.5 VTS 双机热备系统功能模块分析

由以上的分析设计可以看出，VTS 双机热备系统需要对服务器节点的资源和应用程序的进程进行监控和处理，如果发现系统出现故障，则做相应的处理（如切换、发出故障信息等），并具有相应的心跳检测机制和故障处理机制，使用心跳检测机制对系统资源（比如内存、CPU、磁盘、网络接口、网络连接等）进行监测、对应用程序的工作状态进行监测，如果发现系统资源失效、应用程序出现故

障，则按照预先定义好的故障处理机制对系统进行故障处理。其故障处理方式一般都是首先将出现故障的节点从系统中清除并将主服务器节点的工作转移到备份服务器节点，等出现故障的节点恢复以后再将该节点重新纳入系统中，重新参与 VTS 双机热备系统的运行。

所以为了完成上述的设计要求，本文采用双机软件来实现 VTS 双机热备系统中双机管理层的主要功能。双机软件主要有以下几个功能模块组成：中心管理模块、监控模块、执行模块、配置模块，如图 3.9。

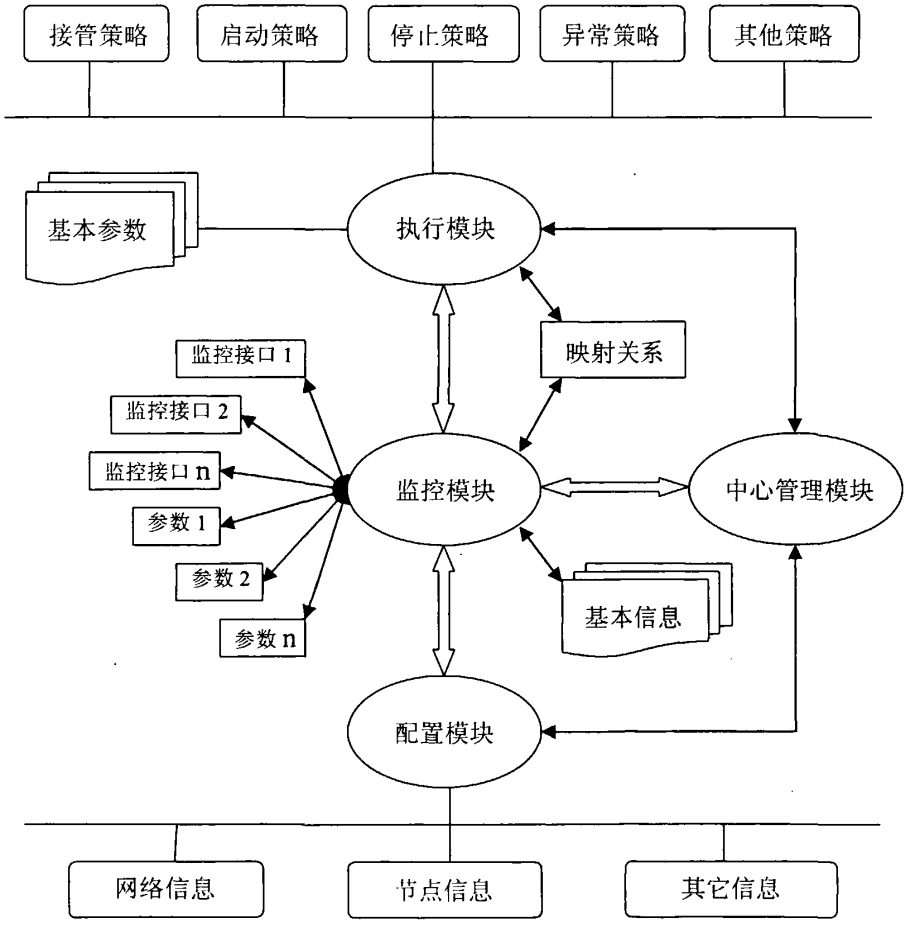

图 3.9 双机软件体系结构

Fig. 3.9 The architecture of dual-machine software

中心管理模块的功能主要是使双机软件与系统资源协同工作的同时，将监控模块、执行模块、配置模块的信息及任务进行分配管理，就好比双机热备系统的“大脑”，支配着其它“部位”协调工作。监控模块的功能是对要监控的应用程序提供监控接口实行检测，它可以提供多个监控接口，即可以监控多个渠道的心跳信息。监控模块如果发现该应用程序不存在，则上报给中心管理模块，中心管理模块将会根据接收到的数据进行故障诊断，并发出指示命令给执行模块。执行模块的功能是根据中心管理模块发出的指令而执行相应的策略，如接管策略等。配置模块的功能是将双机热备系统两个服务器节点的信息汇报给中心管理模块，并且定时更新，对系统的状态及时判断。

第 4 章 VTS 双机热备系统的可靠性

一直以来，人们对提高计算机系统的可靠性问题进行了一系列的探索和尝试，尽可能加强系统的功能以减少危险，从而提高了系统的可靠性。基于前面章节中设计的原理和结构，本章将对海事局 VTS 双机热备系统的可靠性进行研究。利用相关分析模型对 VTS 双机热备系统来进行建模分析，从而对提高系统的可靠性提出改进方法。

4.1 可靠性相关理论

可靠性指零件、机器或系统，在规定的工作环境下，在规定的时间内具有正常工作性能的能力。狭义可靠性指一次性使用的机器、零件或系统的使用寿命[31]。

人们通常用可靠度、MTTF 及故障率来衡量可靠性的指标：

可靠度(Reliability)：指机器、零件或是系统从开始工作起，在规定条件下的工作周期内，达到所规定的性能，即处于无故障的正常工作状态的概率，用 R(t)表示。它定义为：系统在 T_1 时刻正常工作的条件下，在$\{T_1, T_2\}$时间区间内正常工作的概率。它是规定时间 t 的函数，一般来说时间越长，R(t)越小。

平均无故障工作时间 MTTF(Mean Time To Failures)[32]：也可译为平均失效前时间，指可修复的零件机器、或是系统，相邻故障之间的平均正常工作时间，也就是表示系统能够连续提供服务的能力，MTTF 的长短，通常与使用周期中的产品有关，但不包括老化失效。平均修复时间 MTTR(Mean Time To Repair)：也可译为平均恢复前时间，它是随机变量恢复时间得期望值。指用于修复系统和在修复后将它恢复到工作状态所用的平均时间。它包括确认失效发生所必需的时间，以及维护所需要的时间。MTTR 也必须包含获得配件的时间，维修团队的响应时间，记录所有任务的时间，还有将设备重新投入使用的时间。MTTR 是表示一个系统可维护性(Serviceability)参数，也就是修复系统故障使系统恢复正常的能力，MTTF 和 MTTR 的关系如图 4.1。

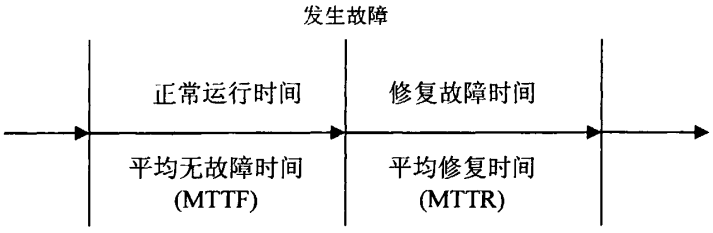

图 4.1 MTTF 和 MTTR 示意图

Fig. 4.1The map of MTTF and MTTR

故障率：通常指瞬时故障率，又称失效率、风险率。即指能工作到某个时间的机器、零件或系统，在连续单位时间内发生故障的比例，即某种工作设备在t时间后的单位时间内发生故障的台数相对于t时间内参与工作的台数的百分比值，通常用λ表示。

由此就可以将 VTS 双机热备系统的可靠性定义为：在给定的时间内，双机热备系统能实施应有功能或提供正常应用服务的能力。

由于双机热备系统由硬件和软件组成，它们对整个系统的可靠性影响呈现完全不同的特性，所以，双机热备系统的可靠性是指分别研究硬件的可靠性和软件的可靠性。

硬件故障主要和零部件制造工艺、组装质量、自然损耗、易维护性有关。它和产品设计有关系但不直接。硬件的可靠性度量在计算机界比较统一，用平均两次故障相隔时间度量。如一台机器每 78 小时左右出一次故障，另一台 200 小时左右，则后者比前者可靠。

软件故障表现为程序计算结果有时正确有时不正确或者某些功能的缺乏。例如，某些输入组常常出错，其余的则没有问题。这些缺陷的原因往往可追溯到软件设计上，是软件的内在缺陷。如果能够排除则软件可靠性增加。但往往排除了一个缺陷又引发了另外几个潜藏的缺陷，这就引起可靠性降低。

4.2 可靠性分析模型的选取

在系统可靠性的研究方法中，主要有组合模型、动态故障树模型、神经网络

模型和 Markov 模型等分析方法。这四种模型中有各有各的特点：1、组合模型在比较复杂的系统很难分析，它比较适合解决简单的静态系统的可靠性问题，所以一般时候不采用这种模型；2、动态故障树模型在分析系统的薄弱环节方面有着较大的优势，但对具有随机性的系统和顺序相关的系统来说，有很大的复杂度；3、神经网络模型在系统的设计过程中，对系统参数的选择起指导作用，对评价既定系统的可靠性还需要进一步的研究和探索；4、Markov 模型不但易于理解而且建模相对简单，而且很适合描述分析具有计算机容错系统的状态转移过程。

由于 VTS 双机热备系统中的状态转换符合 Markov 模型，所以本文选择 Markov 模型来对 VTS 双机热备系统进行建模分析。

4.3 Markov 模型建模思想

系统或（过程）在时刻 t_0 所处的状态为已知的条件下，系统在时刻 $t>t_0$ 所处状态的条件分布与过程在时刻 t_0 之前所处的状态无关的特性称为 Markov 性或者无后效性[33]。可以理解为系统的“将来”的情况与“过去”的情况是无关的。而具有 Markov 性的随机过程称为 Markov 过程，用分布函数表示为：

设 I：随机过程{X(t), t∈T}的状态空间，如果对时间 t 的任意 n 个数值，$t_1<t_2<\cdots\cdots<t_n, n\geq 3$，$t_i\in T$，

恰有 $P\{X(t_n)\leq x_n \mid X(t_1)=x_1, X(t_2)=x_2, \cdots\cdots, X(t_{n-1})=x_{n-1}\}$

$= P\{X(t_n)\leq x_n \mid X(t_{n-1})=x_{n-1}\}$ $x_n\in R$

时间和状态都是离散的 Markov 过程称为 Markov 链。简记为$\{X_n=X(n), n=0,1,2,3\cdots\cdots\}$

用分布律来描述 Markov 性，对于任意的正整数 n，r 和

$0\leq t_1<t_2<\cdots< t_r<m$；$t_i$，m，$m+n\in T_i$ ，

有 $P\{X_{m+n}=A_j \mid Xt_1=Ai_1, Xt_2=Ai_2, \cdots\cdots, Xt_r=Ai_r, X_m=A_i\}$

$= P\{X_{m+n}=A_j \mid X_m=A_i\}$ 其中 $A_i\in I$

称条件概率 $P_{ij}(m,m+n)= P\{X_{m+n}=A_j \mid X_m=A_i\}$ 为 Markov 链在时刻 m 处于状态 A_i 条件下，在时刻 m+n 转移到状态 A_j 的转移概率。由转移概率组成的矩阵称为

Markov 链的转移矩阵。

Markov 模型涉及的转移概率是能否进行准确预测的关键。转移概率矩阵具有的特点：

$$\sum_{j=1}^{\infty} P_{ij}(m,m+n)=1, i=1,2,3,\cdots \tag{4.1}$$

由此可见此矩阵的每一行之和等于 1，矩阵中各元素具有非负性。

当转移概率 $P_{ij}(m, m+n)$只与状态 i，j 以及时间间距 n 有关时，称转移概率具有平稳性。同时也称马尔可夫链是齐次的或者时齐的。此时，记：$P_{ij}(m,m+n)=P_{ij}(n)$,$P_{ij}(n)=P\{X_{m+n}=A_j \mid X_m=A_i\}$,它称为 Markov 链的 n 步转移概率矩阵，当 n=1 时，$p_{ij}=P_{ij}(1)=P\{X_{m+1}=A_j \mid X_m=A_i\}$为一步转移概率矩阵。

综合以上可知，Markov 模型是研究某一事件的状态及状态之间转移规律的随机过程。由于它的基本特征就是“无后效性”,即状态转移概率仅与转移出发状态、转移步数、转移后状态有关,而与转移前的初始时刻无关。根据 t 时刻的状态就可预测 t+△t 时刻的状态，从而制定改善系统的策略。这就是应用 Markov 模型对 VTS 双机热备系统可靠性建模研究的基本思想。

4.4 VTS 双机热备系统数学模型

4.4.1 系统的 Markov 过程

在 VTS 双机热备系统中，系统总共有四种情况：服务器 A 工作（B 作备机）、服务器 B 工作（A 作备机）、单机工作、双机失效。

为了能正确地反应双机热备系统服务器节点在工作状态和失效状态之间的不断转换，在 Markov 模型中定义了所有可能的节点状态和状态转移[34]。其中节点状态描述了在该系统中的任何时刻节点可能处于的状态，正如前一节中所阐述的 Markov 模型定义中，系统的下一步运行状态和如何进入当前状态无关，而仅仅与当前状态有关系。状态转移表示了节点从一个状态转移到另外一个状态的概率[35]。

由此可以构建 VTS 双机热备系统的 Markov 模型，根据系统工作情况定义系统的 Markov 状态集，S={0，1，2，…，n}，如图 4.2，共有 3 个状态：

状态 0：系统两个服务器节点均无故障，工作状态良好。

状态 1：成功检测到系统主节点或备用节点有一个发生故障，并进行了切换或故障排除，系统处于单节点运行的工作状态。

状态 2：系统处于双机失效即危险状态。

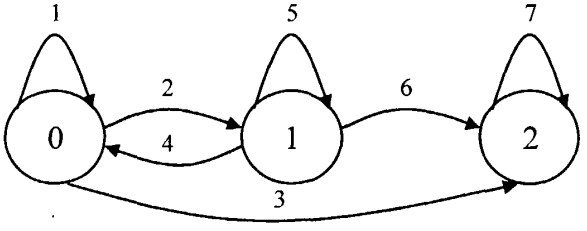

图 4.2 系统状态转换图

Fig. 4.2 The transition of system status

根据 MTTF 的概念，我们可以看出，对于提高 VTS 双机热备份系统的可靠性，基本上有两种方法：增加 MTTF 或减少 MTTR。增加 MTTF 要求增加系统的可靠性；而对于系统而言，当故障的产生难以进行有效的预测和消除时，可以通过快速故障检测维修，降低平均修复时间(MTTR)，从而达到提高可靠性的目的。

所以这里对 VTS 双机热备系统作如下假设：

(1) 系统只能取离散的状态，并且系统在这些离散状态之间转换。

(2) 不考虑状态切换的成功率及时间对系统可用度的影响；

(3) 状态转移可能在任意时刻发生，但是在相当小的时间段内，不可能发生两次状态转移过程。

(4) 系统的故障是独立统计的，服从指数分布，λ表示双机热备系统的故障率；一般故障率取 10^{-5}/h，即%/10^{3}h[36][37]。

(5) 设系统的检测率θ（主要指软件方面的应用程序错误检测以及硬件方面的故障检测的概率）据资料统计，一般来讲系统成功检测到故障并加以修复的概率大约是 80%左右[38][39]。

通过以上的定义，假设系统在时刻 t 正常工作，那么系统在 t+△t 时刻的状态就可以根据指数分布的性质(将△t 看成无限小)得出：时刻 t+△t 失效的概率 w 为：$w=1-e^{-\lambda\Delta t}$，如果把上面部分用指数展开，根据极限的性质，当△t 很小时，$w=1-e^{-\lambda\Delta t}=\lambda\Delta t$。

由此我们可以分析系统状态之间的转移概率（对应图 4.2），具体如下：

1、系统经过一段时间后两台服务器都没有发生故障，

即：$P_{0,0}(\Delta t)=P\{X(t+\Delta t)=x_0|\ X(t)=x_0\}=(1-\lambda\Delta t)(1-\lambda\Delta t)=1-2\lambda\Delta t+o(\Delta t)$

所以系统经过△t 后仍然处于状态 0 的转移概率是 $1-2\lambda\Delta t$。

2、如果在初始状态下经过一段时间后该系统某个节点出现故障，

即：$P_{0,1}(\Delta t)=P\{X(t+\Delta t)=x_1|\ X(t)=x_0\}=C_2^1(1-\lambda\Delta t)\lambda\Delta t=2\lambda\Delta t+o(\Delta t)$

所以系统由状态 0 到状态 1 的转换概率是 $2\lambda\Delta t$，

3、如果在初始状态下经过一段时间后该系统的两个节点均出现故障，

即：$P_{0,2}(\Delta t)=P\{X(t+\Delta t)=x_2|\ X(t)=x_0\}=\lambda\Delta t\ \lambda\Delta t=o(\Delta t)$

所以系统由状态 0 到状态 2 的转移概率为 0。

4、系统在单节点的运行状态下，成功将故障节点检测并加以修复，

即：$P_{1,0}(\Delta t)=P\{X(t+\Delta t)=x_0|\ X(t)=x_1\}=(1-\lambda\Delta t)\theta\Delta t=\theta\Delta t+o(\Delta t)$

所以系统由状态 1 到状态 0 的转移概率为 $\theta\Delta t$。

5、系统仍然处于单节点运行状态的概率为该运行节点既没有发生故障，而故障节点也没有检测到，

即：$P_{1,1}(\Delta t)=P\{X(t+\Delta t)=x_1|\ X(t)=x_1\}=(1-\lambda\Delta t)(1-\theta\Delta t)=1-(\lambda+\theta)\Delta t+o(\Delta t)$

所以系统经过△t 后仍然处于状态 1 的转移概率为 $1-(\lambda+\theta)\Delta t$。

6、若系统在单节点运行的状态下，正常运行的节点也出现故障，

即：$P_{1,2}(\Delta t)=P\{X(t+\Delta t)=x_2|\ X(t)=x_1\}=(1-\theta\Delta t)\lambda\Delta t=\lambda\Delta t+o(\Delta t)$

所以系统由状态 1 到状态 2，其转移概率为 $\lambda\Delta t$。

7、由于计算的是系统的平均无故障工作时间即 MTTF 时，当系统达到状态 2 时，系统已经不能继续工作，即便日后检测修复，恢复过程也属于平均修复时间即

MTTR。所以视为状态 2 为死循环状态，不具有可靠性。

由以上分析可以得到在 Markov 模型中的转移概率矩阵(表 4.1)。

表 4.1 系统工作状态转移概率矩阵

Tab. 4.1 The transfer probability matrix of the system working status

T\T+△T	状态 0	状态 1	状态 2
状态 0	$1-2\lambda\Delta t$	$2\lambda\Delta t$	0
状态 1	$\theta\Delta t$	$1-\lambda\Delta t-\theta\Delta t$	$\lambda\Delta t$
状态 2	0	0	1

4.4.2 模型方程建立与求解

如果用 $P_n(t)$表示系统在时刻 t 处于状态 n 的概率，$P_n(t+\Delta t)$表示系统在 $t+\Delta t$ 时刻处于状态 n 的概率，其中 n 的取值是{0，1，2}中的一个。可以得到如下方程组：

$$\begin{cases} P_0(t+\Delta t)=(1-2\lambda\Delta t)P_0(t)+\theta P_1(t)\Delta t \\ P_1(t+\Delta t)=2\lambda P_0(t)\Delta t+(1-\lambda\Delta t-\theta\Delta t)P_1(t) \\ P_2(t+\Delta t)=\lambda P_1(t)\Delta t+P_2(t) \end{cases} \tag{4.2}$$

变换上面的方程组，移项得：

$$\begin{cases} \dfrac{P_0(t+\Delta t)-P_0(t)}{\Delta t}=-2\lambda P_0(t)+\theta P_1(t) \\ \dfrac{P_1(t+\Delta t)-P_1(t)}{\Delta t}=2\lambda P_0(t)-(\lambda+\theta)P_1(t) \\ \dfrac{P_2(t+\Delta t)-P_2(t)}{\Delta t}=\lambda P_1(t) \end{cases} \tag{4.3}$$

当$\Delta t\to 0$ 时，由导数的定义可得下列微分方程组：

$$\begin{cases} P_0'(t)=-2\lambda P_0(t)+\theta P_1(t) \\ P_1'(t)=2\lambda P_0(t)-(\lambda+\theta)P_1(t) \\ P_2'(t)=\lambda P_1(t) \end{cases} \tag{4.4}$$

由 VTS 双机热备系统的初始条件，当系统处于 t=0 的时刻时，系统完全可靠，

没有故障发生，系统两个节点处于无故障工作状态，故 $P_0(0)=1$，$P_1(0)=P_2(0)=0$。由此求解上面的微分方程组，得：

$$P_0(t)=\frac{(c_1+\lambda+\theta)e^{c_1t}}{c_1-c_2}-\frac{(c_2+\lambda+\theta)e^{c_2t}}{c_1-c_2}$$

$$P_1(t)=\frac{2\lambda e^{c_1t}}{c_1-c_2}-\frac{2\lambda e^{c_2t}}{c_1-c_2}$$

其中：$c_1=\frac{-(3\lambda+2\theta)+\sqrt{\lambda^2+4\lambda\theta}}{2}$

$$c_2=\frac{-(3\lambda+2\theta)-\sqrt{\lambda^2+4\lambda\theta}}{2}$$

根据VTS双机热备系统可靠性的定义，系统能够正常提供服务的状态是在S=0或者1时，所以只求在状态0、1的概率。于是得到系统的可靠度：

$$R(t)=P_0(t)+P_1(t)=\frac{(c_1e^{c_1t}-c_2e^{c_2t})+(3\lambda+\theta)(e^{c_1t}-e^{c_2t})}{c_1-c_2} \tag{4.5}$$

4.5 模型系统可靠性的分析

4.5.1 MATLAB 解析数学模型

通过上面求出的方程解，可以利用 MATLAB 来进行数值分析计算，从而对影响系统性能的关键参数进行评估。本文为了便于比较，当评估某一参数时，另一个参数的值均是由查阅相关资料或根据实际情况等提出的平均假设值。

用 MATLAB6.5 中的微分方程编辑器(Differential Equation Editor)来进行数值分析[40]，操作大体过程如下：

(1) 在 MATLAB 的命令窗口输入 dee 命令，回车，在 command windows 下输入 simulink，在弹出的窗口中选择左侧选择 Simulink 下的 Sink，选择右侧窗口中的 To Workspace 图标以及 Source 下的 clock。

(2) 选择 File/New/Model，弹出名为 untitled 窗口，将 Differential Equation Editor 和 XY Graph 拖入此窗口中。

(3) 双击 dee 窗口中的 Differential Equation Editor 弹出的 dee/DEE 编辑窗口，在 Name 栏中输入模型名称：Markov，在“dx/dt”栏中输入模型，如图 4.3（在计算

时带入参数的具体值，这里的 x(1)代表 $P_0(t)$，x(2)代表 $P_1(t)$)，“x0”栏中输入初始值，“y=”栏中输入要输出的变量。

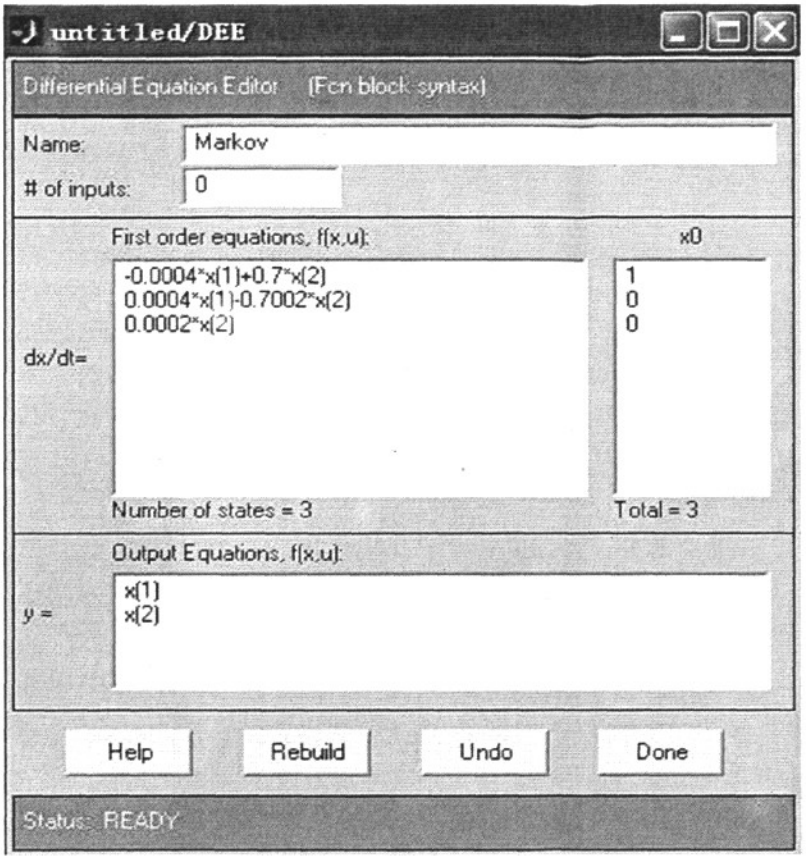

图 4.3 模型方程

Fig. 4.3 Model equations

(4) 点击 Done 回到 untilted 窗口，在 Differential Equation Editor 拖拽箭头，把运行结果输入到相关的工作空间，如图 4.4：

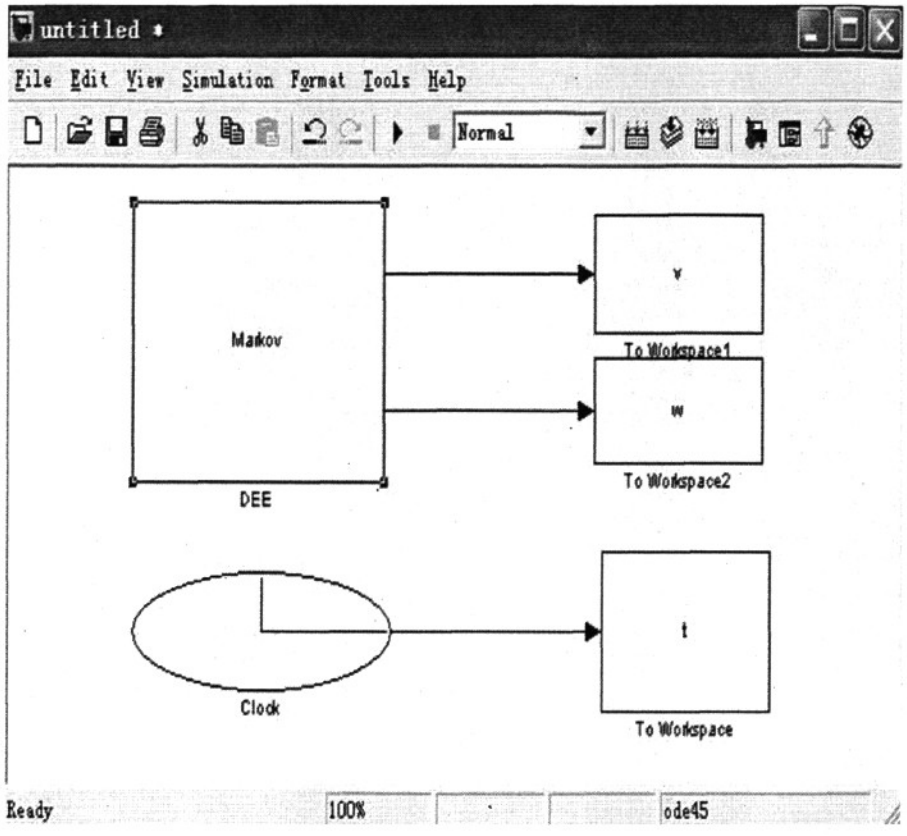

图 4.4 数值仿真

Fig. 4.4 Numerical simulation

(5) 回到 dee 窗口，点选 Simulation 下的 Simulation Parameters 选项，来设置数值参数，在对话框的 Solver 标签页中设置时间，求解器等，在 Workspace I/O 标签页中设置要输入到 MATLAB 工作窗口中的变量。然后选择 Simulation 中的 Start 就可以开始数值计算。在 workspace 的窗口中双击 xout 就可以看到弹出的 Array Editor: xout 窗口，用来显示仿真的数值。

4.5.2 相关参数的讨论与分析

当对检测修复率 θ 不变，故障率变化时，进行数值分析，得出模型的可靠度数据，整理后如表 4.2 所示。

表 4.2 可靠度数据表 λ 变化

Tab. 4.2 Reliability data sheet λ changed

时间 t	λ=0.0002 θ=0.7	λ=0.0001 θ=0.7
单位 h	R(t)	R(t)
0	1	1
500	0.9903	0.9926
1000	0.9829	0.9887
2000	0.9613	0.9772
6000	0.9095	0.9389
9000	0.8647	0.9018
12000	0.8406	0.8801
15000	0.8217	0.8583
20000	0.8126	0.8411

在 MATLAB 的命令窗口中通过 plot 命令来绘制 λ=0.0002 θ=0.7（虚线）和 λ=0.0001 θ=0.7（实线）可靠度的曲线，如图 4.5。

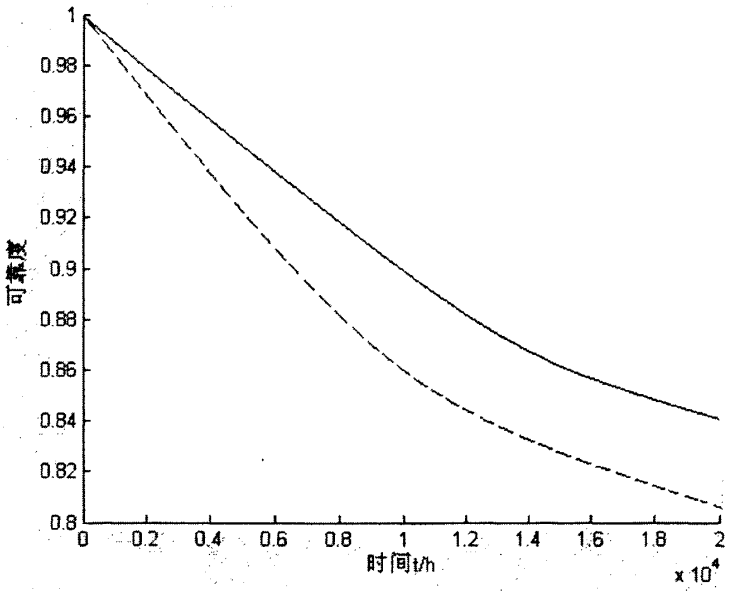

图 4.5 可靠度曲线图

Fig. 4.5 The figure of reliability curve

当对故障率 λ 不变，检测修复率变化时，进行数值分析，得出模型的可靠度数

据，整理后如表 4.3 所示。

表 4.3 可靠度数据表θ变化

Tab.4.3 reliability data sheet θ changed

时间 t 单位 h	θ=0.6 λ=0.0001 R(t)	θ=0.8 λ=0.0001 R(t)
0	1	1
500	0.9903	0.9963
1000	0.9609	0.9881
2000	0.9484	0.9789
6000	0.9079	0.9432
9000	0.8901	0.9211
12000	0.8803	0.9016
15000	0.8662	0.8925
20000	0.8518	0.8807

图 4.6 是θ=0.6 λ=0.0001（虚线）和θ=0.8 λ=0.0001（实线）可靠度的曲线，

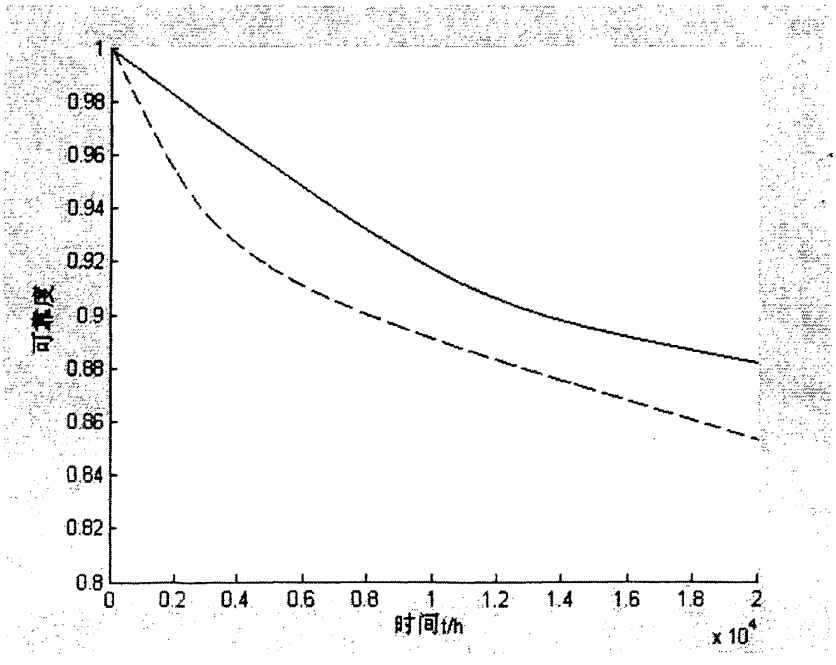

图 4.6 可靠度曲线图

Fig. 4.6 The figure of reliability curve

从上面的分析、计算和绘图可以得出：

(1)双机热备系统的可靠度随着时间的变化而降低。

(2)当系统的故障率是一个常数时，系统的可靠度度随着系统的故障率改变而成反方向变化。当故障率增加时系统的可靠度降低；当系统的故障率比较低的时候，系统的可靠度也相对较高。

(3)当系统的检测率为一个常数时，系统的可靠度随着时间的变化而下降。检测率越小，系统可靠度随时间的变化下降的幅度越剧烈；检测率较大时，可靠度随时间变化而下降的幅度较为缓慢。

所以为了提高双机热备系统的可靠度,需要做下列工作：

①降低双机热备系统的故障率。②提高系统的检测率。③定期对系统中的设施进行检查测试，以较小的维护成本代替较大的维修成本，从而提高系统的可靠性。

另外，正如本章前面所讲，通过延长 MTTF 或降低 MTTR 可以提高双机系统的可靠性。通过 4.3.2 节中的微分方程解，得到平均无故障工作时间 MTTF 为：

$$\text{MTTF}=\int_0^{+\infty} R(t)dt=\int_0^{+\infty}[Q_0(t)+Q_1(t)]dt \tag{4.6}$$

$$=\int_0^{+\infty}\left(\frac{(c_1+3\lambda+\theta)e^{c_1 t}}{c_1-c_2}-\frac{(c_2+3\lambda+\theta)e^{c_2 t}}{c_1-c_2}\right)dt$$

$$=\frac{c_1+3\lambda+\theta}{c_1(c_1-c_2)}e^{c_1 t}\Big|_0^{+\infty}-\frac{c_2+3\lambda+\theta}{c_2(c_1-c_2)}e^{c_2 t}\Big|_0^{+\infty}$$

$$=-\frac{c_1+3\lambda+\theta}{c_1(c_1-c_2)}+\frac{c_2+3\lambda+\theta}{c_2(c_1-c_2)}$$

$$=\frac{3\lambda+\theta}{c_1 c_2}$$

可见，MTTF 的长短与系统故障率和检测率密切相关的。但直接延长 MTTF 受设备制造工艺、传输线路质量等硬件因素影响，难以控制；而可以通过设置冗余设备、链路和快速侦测到故障来缩短 MTTR（从另外的角度来说就是间接的延长 MTTF），从而提高系统的可靠性。

综上所述，要实现具有高可靠性的 VTS 双机热备系统就需要从硬件和软件两个方面来入手，并且也要在实施和维护时需要考虑能使双机热备系统可靠的各种因素。由于服务器节点的故障率（硬件方面来看）基本上是比较固定的，要想降低其故障率比较困难，只能尽量采用一些大品牌的服务器硬件设施。所以要想提高整个 VTS 双机热备系统正常工作的时间，即提高系统可靠性，则必须提高检测率。而对于提高本文的 VTS 双机热备系统的检测率，可以从系统的故障侦测方面来入手，通过改进双机热备系统的某些功能来提高检测率，从而达到提高系统可靠性的目的。

4.5.3 影响系统可靠性的因素

1、应用程序检测

从研究分析可以看出，系统中采用的双机软件同时对系统资源和应用程序进行监测和处理。它们监测和处理应用程序的大致如下[41]：①双机软件将应用程序的路径记录到它的监控列表中；②应用程序的启动是根据双机软件记录的应用程序路径来启动；③对该应用程序的进程进行监控是双机软件利用心跳机制来完成；④当应用程序的进程不存在被检测到时，双机软件执行故障处理。

所以，在对应用程序的监控方面，双机软件只能判断应用程序进程是否存在，不能对应用进程的其他状态进行判断。因为只有当应用程序的进程丢失时，系统才能切换。然而进程存在并不表示该进程的工作是正常的，与进程相关的服务很有可能出现了异常，如运行不稳定或程序总是出错等，很显然，这将导致系统的故障检测率降低，因为应用程序可能已经发生了错误，但是双机软件并不能进行判断，系统的可靠性将降低。

2、心跳检测

VTS 双机热备系统的心跳检测用于判断对方节点系统资源的状态，并作为切换的依据。系统提供了两种心跳的通信方式。(1) 内网连接。能够支持 TCP/IP 的通讯协议，例如：以太网等。(2) 串行线 。串口通信方式需要利用 RS232 的拟调解线路来将双机热备系统相连接。心跳的通信方式越多，心跳检测的效率就越高，因为多一种心跳检测的方式，心跳检测同时失效的可能性要小很多，从而可以提

高双机热备系统的可靠性。如果心跳检测的效率不高，出现心跳丢失或失效，就会导致在一定的时间范围内，心跳的有效判别数量下降。这样会造成双机热备系统的切换不及时或者无效切换，导致可靠性的降低。

4.6 VTS 双机热备系统的改进

4.6.1 应用程序检测的改进

在 VTS 双机热备系统的层次结构、工作流程设计、模块功能分析以及可靠性方面研究的基础上，通过让双机热备系统能够应用进程的状态进行监控，即加强系统对应用程序的侦测能力，从而就可以提高系统的可靠性。所以本文通过加入脚本文件和对应用进程状态监控模块来提高系统对应用程序的检测率。

1、利用脚本 kill 僵死进程

我们知道，由于网络等的原因，有些进程会突然僵死。这些进程已经死亡，但没有释放系统资源，包括内存和一些系统表等，如果这样的进程很多，会消耗系统大量的资源，直接影响服务器的正常运行。为了实时地、自动地杀死这些僵死的进程，可以下面的这个脚本来完成。

```
# kill_zombie
ps -ef | awk '{ print    $1,$2,$7,$8 }'|
awk '/[0-9][0-9]:[0-9][0-9]:[1-9][0-9]/{ print $1,$2,$3,$4}' |
awk '!/root/ { print "kill -9" $2}' > /tmp/ kill_zombie
chmod 777 /tmp/ kill_zombie
/tmp/ kill_zombie
```

首先，用命令 ps -ef 查看进程状态，通过管道传送给 awk 进行处理。在第一个 awk 中，获取进程的用户标识(UID)、进程号(PID)、进程占用 CPU 时间(Time)、进程执行命令(CMD)四个字段的值。在第二个 awk 中，通过模式匹配，选取所有匹配模式的行。在 awk 中，[0-9]匹配 0～9 中任一个数字，[1-9]匹配 1～9 中任何一个数字，连用两个[0-9][0-9]则匹配一个任意两位的数字，因此[0-9][0-9]:[0-9][0-9]:[1-9][0-9]则匹配 Time 时间字段值，查找占用 CPU 时间超过 10 秒的进程；如果要查找占用 CPU 时间超过半小时的进程，则把模式改成

[0-9][0-9]:[3-9][0-9]:[0-9][0-9]。在第三个 awk 中，用"!/root/"过滤掉 root 用户生成的进程，并进行 Shell 语言拼装，并将最终结果定向到文件/tmp/kill_zombie。在/tmp/kill_zombie 文件中，都是形如 kill -9 123 的 Shell 命令。最后，执行/tmp/kill_zombie 杀死进程。最后用 crontab -e 增加一个 cron 作业：*/3 * * * * /tmp/kill_zombie

2、进程状态监控模块

该模块的功能是如果该应用进程的状态不是正常运行时的状态，那么该模块会 kill 掉进程或者让双机软件进行判别并执行相关操作，如图 4.7，它是通过监控接口与监控模块进行通信，在切换时仍然通过中心管理模块调用执行模块来完成。

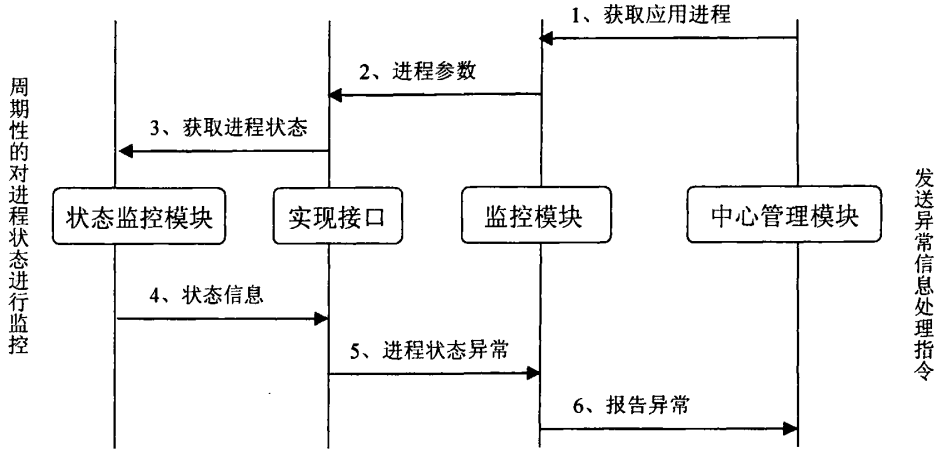

图 4.7 状态监控表述图

Fig. 4.7 The statement map of the monitoring status

经研究分析，一般应用程序进程陷入死循环这种情况很少出现，大多数的情况是：进程出现问题，都会因为非法请求资源被挂起或者出现异常。如果可以排除网络的原因，从进程的状态就能判断出服务是否正常。这种方案适用于需要检测的应用服务，对于其进程状态进行监控，下面是该模块中主要函数的功能和部分代码说明。

(1) 应用进程信息的获取

通常我们查看某一进程状态的时候用 ps 命令，而 ps 是通过对/proc 文件的访问来实现的[42]。/proc 文件系统是一种内核和内核模块用来向进程(process)发送信息的机制。它可以用于获取运行中进程的信息。在/proc 中有一些编号的子目录。每个编号的目录对应一个进程 id(PID)。这样，每一个运行中的进程/proc 中都有一个用它的 PID 命名的目录。相关函数 get_process()的部分代码如下：

```
……..
struct dirent *pdir;
char filename[255];
pd=opendir("/proc");
while((pdir=readdir(pd)!=NULL)){ //循环查看进程文件，并调用
    if(pdir->d_ino==0)              //监控进程函数
        continue;
    strcpy(filename,"/proc/");
    strcat(filename,pdir->d_name);
    if((fd=open(filename,"r")!=-1)
        monitor_process();//调用进程监控函数 or something
}
……..
```

(2) 应用进程状态的监控

进程的状态一般有以下几种：D 不可中断 uninterruptible sleep (usually IO) 收到信号不唤醒和不可运行, 进程必须等待直到有中断发生; R 运行 runnable (on run queue)正在运行或在运行队列中等待; S 中断 sleeping 休眠中, 受阻, 在等待某个条件的形成或接受到信号; T 停止 traced or stopped 进程收到 SIGSTOP, SIGSTP, SIGTIN, SIGTOU 信号后停止运行; Z 僵死 a defunct ("zombie") process 进程已终止，但进程描述符存在。该函数通过读取系统的/proc 文件，将其内容输入到一个文件中，通过读取该文件，来判断进程的状态，并作出相应的处理。相关函数 monitor_process()的部分代码如下：

```
.....
printf(" %d\n " ,ppid);
char *filename = (char *)malloc(100);
sprintf(filename, "/proc/%d/status ",ppid);
FILE *pFile;
char *line=null;
char *add;
int pronum; /*需要监控的应用进程行号*/
pFile = fopen (filename, "r" );
if (pFile == null){ /*打开文件是否成功*/
    printf(" %s\n " , "config file is not exist!\n ");
    return -1 ;
}
int linenum = 0 ;
char* tmp;
while ((add = fgets(&line, 100, pFile))!= null &&linenum < n ){
    linenum ++ ;
    strtok(line, "\t" ); //分割字符串
    tmp = strtok(NULL, "....." ); //将表示状态的字符串赋值给 tmp
    if (linenum==pronum &&((strcmp(tmp, "S")==0)||(strcmp(tmp,"R")==0))){
                    //判断应用进程是否是正常状态，否则做相应处理
        ...../关闭文件，返回等操作
}
else   send_message();//or kill_process()调用相关函数处理
.....
```

(3) kill 或重启应用进程

在 Linux 系统中进程是一个非常重要的概念。Linux 是一个多任务的操作系统，系统上经常同时运行着多个进程。kill 不关心这些进程究竟是如何分配的，或者是内核如何管理分配时间片的，所关心的是如何去控制这些进程，让它们能够很好

地为用户服务。在进程管理方面 Linux 用分时管理方法使所有的任务共同分享系统资源。主要有批处理进程，交互进程，监控进程等不同作用的进程类型，并且具备多种启动方式。另外 Linux 具备 Windows 不具有的特点，任务抢占机制，在一些应用程序出现问题的时候可以快速的被系统取代。相关函数 kill_process()的部分代码如下：

```
.....
if (kill(ppid,SIGKILL) == 0 ){
    fd = fork();//判断是否创建了子进程
    if (fd == 0 ){
        execl(const char *path,const char *arg,null);
                //运行其它程序并使用可变参数
    }
    else   /*提示并返回*/
}
.....
```

(4) 消息的传递

模块之间信息的传递过程：1)进程状态监控模块 A 通过实现接口将信息送到内存区块中的缓冲区内，该缓冲区对模块 A 来说是只写(Write Only)的属性；2)双机软件监控模块 B 经由实现来接口接收内存区块中缓冲区的信息，该缓冲区对模块 B 来说是只读(Read Only)的属性；3)模块 B 根据数据信息经由中心管理模块进行相应处理。相关函数 send_message()的部分代码如下：

```
.....
#define  MAX_BUF_SIZE 1024
static char buf[MAX_BUF_SIZE] = {.....}; /*设置缓冲区*/
int mes;
int size; /*相关变量参数*/
mes = send(size, buf, strlen(buf), 0); /*向缓冲区写入数据*/
while(...../设置循环条件){
```

```
        if (mess == -1) {
            pintf("send message failed");/*信息数据为空，发送失败*/
            return -1;
        }
        else
            mes += send(size, buf, MAX_BUF_SIZE, 0);
        }
    setbuf(stdout,buf); /*把缓冲区与流相连*/
    fflush(stdout); /*清除缓冲区*/
    ……..
```

该程序模块编译后可以利用脚本来启动[43][44]，也利用 crontab 增加一个作业来设置程序执行的周期。但是上述方案会有一定的缺陷，本文没有考虑该方法占用的系统资源，如果占用的较多，会导致切换的速度会有所减慢。但是提高了故障检测率，从系统的可靠性方面来讲，还是有了一定的提高。

4.6.2 心跳检测的改进

本文通过在磁盘阵列柜上为双机软件开辟一个分区，此分区用来存放服务器节点的信息，如图 4.8。主备服务器节点通过此区域的数据来判断对方的状态，以此来作为另一种心跳检测的方式。

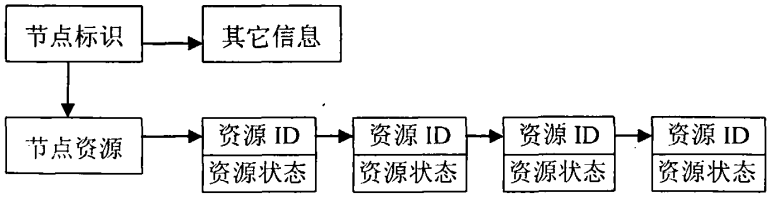

图 4.8 节点信息

Fig. 4.8 Information of node

当前两种心跳方式故障或者由于网络等原因出现较长时间的延迟时，可以通过此种方式来进行故障判别，从而提高心跳检测的效率，进行及时的切换或者避

免不必要的切换，如图 4.9。

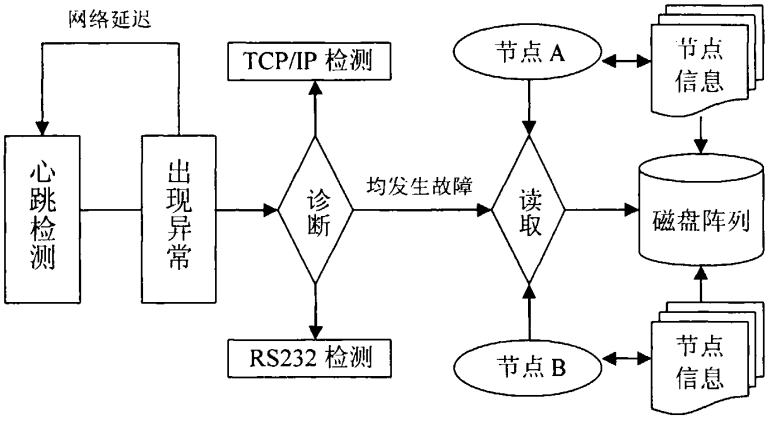

图 4.9 增加的心跳检测方式

Fig. 4.9 The increase way of Heartbeat detection

利用两个节点的双机软件对磁盘阵列中的节点信息进行交换，加强双机热备系统的故障侦测能力，从而达到提高可靠性的目的。另外，在保持硬件平台不变的情况下，降低 MTTR，提高双机热备系统的可靠性也在于快速故障检测即如何有效的缩短发现对方服务器节点失效的时间，这就涉及到设置心跳检测频率（心跳周期）的问题。在一般情况下，心跳检测的频率越高，故障的检测率越高，修复系统的成功率也就越高。但是心跳检测频率设置是一个两难的选择，采用短心跳周期的系统 MTTR 也较短，但这样系统监测的负担就比较大。采用长心跳周期会使主服务器宕机时，备份服务器的响应会较慢。这里可以根据应用服务和 VTS 双机热备系统运行的实际情况，通过更改双机软件的配置文件来设置心跳检测的频率。

第 5 章 VTS 双机热备系统的应用

通过对 VTS 双机热备系统的分析设计，以及采用 Markov 预测法对系统的可靠性进行研究，分析了影响系统可靠性的因素，从而提出改进的方案。本章将较为详细的阐述 VTS 双机热备系统具体实现和部署过程，并进行相关的测试。最后通过比较验证系统可靠性的提高。

5.1 VTS 双机热备系统配置

根据海事局 VTS 系统的业务需求和系统的 Markov 数学模型分析结果，本文对系统的硬件和软件两个方面进行了综合考虑。配置如下：

5.1.1 硬件配置

硬件配置采用的是 Dell PowerEdge 2950 Systems 架式服务器两台，服务器操作系统硬盘两块；服务器网卡：四块。Dell 液晶显示器一台；鼠标、键盘各一个；网线若干条。系统中的每一个服务器都由主板、内存、CPU、硬盘、网卡等基本组件构成，而其中的主板、内存、CPU、硬盘、网卡均可通过同型号、同接口的备件更换，基本做到“即插即用”，无需其它软件安装和过多的硬件设置。当检测到系统硬件故障时，可以快速的替换或者维修，从而可以提高双机热备系统的可靠性。

存储设备采用的 Dell PowerVault MD3000，它是架装式外部独立磁盘冗余阵列存储设备，采用 RAID5 技术规范，其专为高可靠性而设计，提供了对数据存储的冗余访问。并且具有良好的故障检测功能，大大提高了系统的检测修复率。另外此种设备还具有 RAID5 磁盘阵列提供的专用校验功能，对数据存储带来了很好的容错性。而且对网络而言，这种存储方式可以不占用网络带宽，减少了网络的数据吞吐量。两台服务器在功能上为一个功能单元，在物理上是两个网络节点和网络地址，客户机的命令、操作均向两台服务器发送，由在线服务器响应回答。同时服务器随时监视网络状态，一旦出现链路中断立即重建链路或输出故障信息作为切换依据之一。

5.1.2 软件配置

不同的双机热备系统要采用与其相适应的双机软件来实现。在第三章中已经阐述了采用的可以满足 VTS 双机热备系统设计要求的双机软件主要模块和功能，它是国内联鼎公司较早研制的一款双机软件。同时在选择双机软件时也充分考虑了 VTS 系统使用的数据库及其版本。另外如果有可能，则还要了解清楚：数据量的大小、数据写入的频率等，由于时间和篇幅的限制，本文暂时没有考虑这方面的因素。

此外 VTS 双机热备系统还需要的软件有 Red Hat Enterprise Linux AS 4 操作系统安装盘；JDK；Tomcat5.5；PostgreSQL；三个外网 IP 地址和两个内网 IP 地址，三个主机名，两台双机热备服务器各自有一个外网 IP、一个内网 IP（作为心跳线）以及主机名，VTS 双机热备系统对外还有个虚拟的 IP 地址和主机名；以及各自软件或应用服务器运行所需要的端口号。总体配置结构如图 5.1。

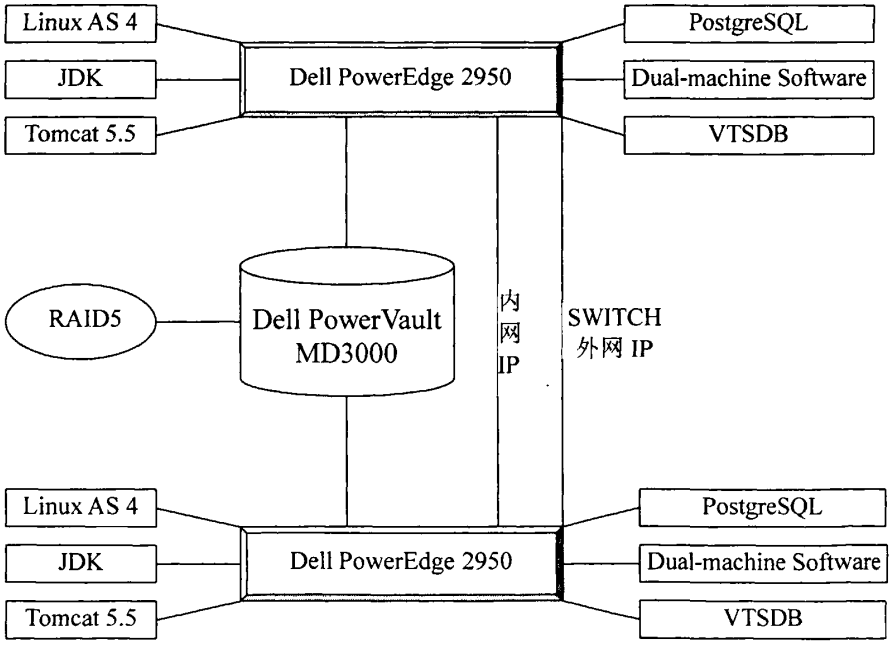

图 5.1 VTS 双机热备系统配置

Fig. 5.1 VTS dual-machine hot standby system setting

5.2 VTS 双机热备系统的部署实现

5.2.1 操作系统层

众所周知，Linux 系统的设计结构使其具有很高的可靠性。另外 Linux 是开放源代码的操作系统，除了成本非常低廉，它的可维护性、可扩展性以及较低的系统占用率和其它操作系统比较而言，都有着较大的优势。因此系统的服务器采用 Red Hat Enterprise Linux Advanced Server 4，内核版本为 Kernel 2.4.21。系统具体的分区、格式、大小以及节点服务器设置相关 IP 属性等如表 5.1、表 5.2。

表 5.1 系统分区表

Tab. 5.1 The table of system partition

目录	格式	大小
/usr	ext3	15000
/tmp	ext3	10000
/var	ext3	10000
/home	ext3	10000
	swap	8200
/	ext3	all

表 5.2 网络配置表

Tab. 5.2 The table of network setting

	Node A	Node B
Operating System	Red Hat Enterprise Linux AS 4.0	Red Hat Enterprise Linux AS 4.0
Node Name	misnormal	misredundant
Etho0	172.16.199.57	172.16.199.58
Subnet Mask	255.255.255.224	255.255.255.224
Etho1	192.168.0.2	192.168.0.3
Subnet Mask	255.255.0.0	255.255.0.0
Gateway	172.16.199.33	172.16.199.33
DNS	202.118.80.2	202.118.80.2

装好操作系统之后，就可以在客户端（本文中使用的是 Windows XP）利用 SSH

登陆到服务器上，并将所要用的软件（Linux 版本）上传到服务器硬盘指定的目录文件夹下，如/tmp。

磁盘阵列是双机热备系统的核心，它的可靠性是关键，系统提供的应用服务不丢失的前提是磁盘阵列部分不出故障，一旦磁盘阵列的控制器故障导致设备无法访问，则无论服务器主机有多好的性能和可靠性，都无法阻止系统停止服务。

本文的阵列柜用来存放 VTS 系统的数据库文件，在分区前确认服务器与磁盘阵列柜相连接的情况下，进入服务器 Linux 操作系统的桌面，新建终端，输入“fdisk -l”就可以看到阵列柜，利用帮助命令进行分区，各部分的设置如表 5.3。

表 5.3 磁盘阵列柜分区表

Tab. 5.3 The table of Disk array partition

名称	用途	大小	ID	文件系统类型
sdb1	应用数据	1～33079	83	ext3
sdb2	双机软件	33079～35568	83	ext3（可选无）

双机服务器与磁盘阵列柜互相连接的结构其好处有：

1、硬软结合实现真正意义上的数据与系统分离；

2、对硬件配置要求不高，服务器也可可以采用不同或相差较大的配置；

3、系统切换时间短，平均切换时间不到为 30s；

4、切换过程对应用程序无影响，无需重新启动或登录；

5、系统效率高，因为整个系统中数据读写、管理及容错由磁盘阵列来完成。而系统从服务器故障到纠错处理由双机软件来完成，而这两个都是相对独立的子系统[45]。

另外，双机与磁盘阵列柜互联结构采用内存镜像技术，可以有效的避免由于应用程序自身的缺陷导致系统全部宕机，同时由于所有的数据全部存贮在中置的磁盘阵列柜中，当工作机出现故障时备份机接替工作机，从磁盘阵列中读取数据，所以不会产生数据不同步的问题，由于这种方案不需要网络镜像同步，因此 VTS 双机热备系统的服务器性能要比镜像服务器结构高出很多。

5.2.2 应用系统层

1、WEB 服务配置

VTS 系统的运行需要在 JDK 和 Tomcat，安装之前，在客户端利用 SSH 将 jdk-6u7-linux-i586.bin 和 apache-tomcat-5.5.20.tar.gz 分别上传到/usr/java/文件夹和/usr/local/文件夹下，新建终端到/usr/java/文件夹下输入以下命令

chmod a+x jdk-6u7-linux-i586.bin 更改文件的权限

./jdk-6u7-linux-i586.bin 执行该文件，开始安装

利用 VI 编辑器，设置环境变量[46]，将以下/etc/profile

export JAVA_HOME=/usr/java/jdk1.6.0_07

export PATH=$JAVA_HOME/bin:$PATH

export CLASSPATH=.:$JAVA_HOME/lib/tools.jar:$JAVA_HOME/lib/dt.jar

配置完环境变量后可以利用 java –version 来测试是否配置成功。

同理，新建终端到/usr/local 文件夹下输入以下命令

tar zvxf apache-tomcat-5.5.20.tar.gz //将该压缩包解压

mv apache-tomcat-5.5.20.tar.gz tomcat //更改文件夹的名称

rm apache-tomcat-5.5.20.tar.gz

启动服务脚本为：

/usr/local/tomcat/bin/startup.sh //用来启动 Tomcat 的脚本

终止服务脚本为：

/usr/local/tomcat/bin/shutdown.sh //用来停止 Tomcat 的脚本

安装好 Tomcat 后就可以将开发好的 VTS 系统应用程序布打成 war 包，布置到 webapps 目录下。

2、PostgreSQL

VTS 系统采用的是 PostgreSQL 数据库，它是面向目标的关系数据库系统，具有传统商业数据库系统的所有功能，同时又含有将在下一代 DBMS 系统的使用的增强特性。PostgreSQL 是自由免费的，并且所有源代码都可以获得。在安装 PostgreSQL 之前，在客户端利用 SSH 将 postgresql-8.3.5.tar.gz 上传到/usr/local/文件

夹下，在桌面新建终端，要卸载目前 Linux 系统自带的版本或者以往装过的版本，命令如下：

for i in `rpm -aq | grep postgre`;do rpm -e --nodeps $i;done

然后按以下步骤安装 postgresql-8.3.5：

1) userdel postgres /*删除当前的用户组及用户，添加新的
2) groupdel postgresql 组及用户，并设置密码*/
3) groupadd postgresql
4) useradd –g postgresql postgres
5) passwd postgres //输入“postgres”
6) cd /usr/local
7) tar xvfz postgresql-8.3.5.tar.gz //解压缩文件
8) cd postgresql-8.3.5
9) ./configure
10) gmake //这个过程大概会花费几分钟的时间
11) gmake install //最后一行会显示出建立完成的信息

接着按以下步骤建立数据库：

1) mkdir /SYPIM/scsi
2) vi ~postgres/.bash_profile //把以下的部分添加到配置文件中

   PGLIB=/usr/local/pgsql/lib //配置 PostgreSQL 的环境变量

   PGDATA=/SYPIM/scsi/data

   PATH=$PATH:/usr/local/pgsql/bin

   MANPATH=$MANPATH:/usr/local/pgsql/man

   export PGLIB PGDATA PATH MANPATH //使变量生效
3) mount /dev/sdb1 /SYPIM/scsi //挂载分区
4) cd /SYPIM/scsi
5) mkdir data
6) chown postgres.postgresql /SYPIM/scsi/data
7) su - postgres

8) initdb -D /SYPIM/scsi/data –W 输入“postgres”

9) cd /SYPIM/scsi/data

利用 vi 编辑器更改 postgresql.conf 和 pg_hba.conf 两个配置文件，在 postgresql.conf 里把 listen_addresses='*' 将第一行#号去掉改成*，目的是让任何地址都可以访问，然后在 pg_hba.conf 填入以下内容：

host all all 127.0.0.1 255.255.255.255 md5 //设置访问的类型、地址等

host all all 172.16.199.33 255.255.255.224 md5

启动服务（前提是切换到 postgres 用户下）

postmaster –D /SYPIM/scsi/data > logfile 2>&1 &

终止服务（前提也是切换到 postgres 用户下）

pg_ctl –D /SYPIM/data stop -ms //等待所有的连接退出后关闭

pg_ctl –D /SYPIM/data stop -mf //不等连接退出直接关闭

3、FTP 服务

配置 FTP 服务器的作用是整个海事系统中有个 SYTAR 雷达系统，雷达会将侦测到的数据以.txt 的格式发送到 FTP 上，VTS 系统会将这个文本文件进行解析，然后显示到页面上来。主要用来侦测船舶的位置和当前的天气状况。配置时按以下步骤来进行：

1) useradd sytar

2) passwd sytar //输入“sytar”

3) cd ~sytar

4) mkdir expected

5) mkdir events //创建文件夹

6) vi /etc/vsftpd.user_list

7) add lines //输入 root

8) vi /etc/vsftpd.conf //修改配置文件

9) 更改以下内容

anonymous_enable=NO

local_enable=YES

userlist_files=/etc/vsftpd/user_list_local

userlist_enable=YES

5.2.3 双机管理层

VTS 双机热备系统的双机管理层主要通过双机软件和对其改进来实现的。安装双机软件之前，同样利用 SSH 将安装源文件上传到服务器的硬盘中。以 root 用户登陆操作系统，按照安装程序的提示，安装软件，其过程类似与 Windows 下一般软件的安装，这里不再赘述。安装完成以后，自动会在/etc/下新建了 Cluster 文件夹，执行 startmanager.sh。在弹出的双机软件界面中启动设置向导开始配置。节点数和资源包数，节点数是 2，包数是 1；资源包名：vtsdb；VTS 双机热备系统网络配置拓扑结构如图 5.2。

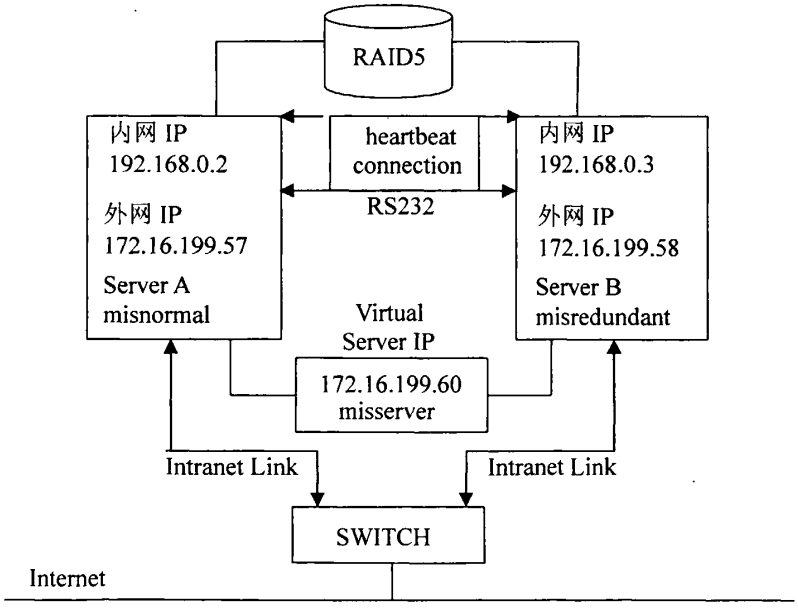

图 5.2 双机热备系统拓扑结构

Fig. 5.2 The topology map of dual-machine hot standby system

心跳 IP 为内网 IP，即 192.168.0.2、192.168.0.3；工作 IP 为外网 IP，即 172.16.199.57、172.16.199.58；Virtual IP 为虚拟 IP，每个包必须有一个虚拟 IP，

这里填写 172.16.199.60；NetMask 为虚拟 IP 的子网掩码，填写 255.255.255.224；虚拟服务器技术是通过改写请求报文的 MAC 地址[47]，将请求发送到真实服务器，而真实服务器将响应直接返回给客户。当客户端访问的虚拟 IP 在主机节点时，备机节点不必使用这个虚拟 IP，当主节点失效时，备份机要将此虚拟 IP 添加到它的网卡上。虚拟 IP 不仅可以使用户的访问不受系统故障时切换的影响，而且还可以一定程度上提高系统的安全性。

在“Process List”中的监视进程必须填写可执行程序开启的所有进程名称，若有多个进程，请用";"隔开，这里填写“java;”和 PostgreSQL 数据库所有的进程名。最后一个进程用";"结尾。

在包属性界面的 Volume 选项卡中“volume list”填写应用数据的分区“/dev/sdb1”。在节点属性的界面“Test Volume”中填写为双机软件开辟的磁盘阵列分区“/dev/sdb2”。

在“Switch Rule”选项中选择切换规则[48]（指定规则/可回切/负载均衡），这里选择“Fall over”(指定规则)。各规则说明如下：

Fall over（指定规则）：按指定的次序进行资源的切换，如图 5.3。正如本文的系统，节点标号分别为 A、B，如果为资源指定的切换次序是 A，B，那么资源将首先在节点 A 启动，当 A 发生故障，A 将自动切换到节点 B，节点 A 的故障修复后，系统不会向回切换，直到节点 B 发生故障。

图 5.3 指定规则示意图

Fig. 5.3 The map of fall over

Fall back（可回切）：总是会向切换次序靠前的可用服务器节点切换。如上例，资源指定的切换次序是 A，B，资源首先在节点 A 启动，当 A 发生故障时，资源

切到节点 B，而节点 A 排除故障恢复正常时，资源又会自动切回到节点 A，也就是主备机的角色不会发生互换。

Balanced load（负载均衡）：此规则是针对双机互备或者双机双工的模式。系统将自动监测每个服务器的包数，将资源多的服务器切到资源少的服务器。如有两台服务器节点 A，B，有两个资源 1、2，初始状态为节点 B 关闭，1、2 均在节点 A 运行，当节点 B 开启后，系统会自动选择一个资源切换到节点 B，使 A、B 两节点各有一个资源任务运行。

为每个服务器节点添加脚本程序，其目的是为了在进行切换时启动或者停止指定的应用服务程序[49]。在“Start Name”项填写启动时的执行文件全路径，这里填写的是/home/start.sh；在“Stop Name”项填写停止时的执行文件全路径，这里填写/home/stop.sh；启动和停止脚本内容如表 5.4。

表 5.4 脚本内容

Tab. 5.4 Script content

脚本名称	脚本内容
start.sh	#bash su - postgres -c "postmaster -D /SYPIM/scsi/data > logfile 2>&1 &" sleep 2 su - root -c "/usr/local/tomcat/bin/startup.sh &" sleep 2 su - root -c "/usr/src/programl/monitor _begin.sh &"
stop.sh	#bash su - root -c "/usr/src/programl/monitor _end.sh &" sleep 2 su - root -c "/usr/local/tomcat/bin/shutdown.sh &" sleep 2 su - postgres -c "pg_ctl -D /SYPIM/scsi/data stop -mf &"

配置完成后，可以看见 Configuration Complete 窗口的状态栏中 synchronizing configuration to every node 进度条，是将其同步到另一个节点上。配置双机软件之后不用重启系统，可以直接利用管理界面启动双机热备后台进程。先执行安装目录中的 CManager，连接到双机热备系统的节点中（连接时需要填入待连接主机的 root 帐户密码）在主界面左边的浏览树中，用鼠标左键选中要操作的节点名，用鼠标右键弹出如下菜单：选择[Start Cluster Service]项，启动指定节点上的后台进程。

停止双机热备服务的操作和启动双机热备服务的操作基本一致。

在实验室的环境下，本文的双机热备正常运行时如图 5.4 所示，现场部署时只需要更改相关 IP 地址、子网掩码以及网关等。

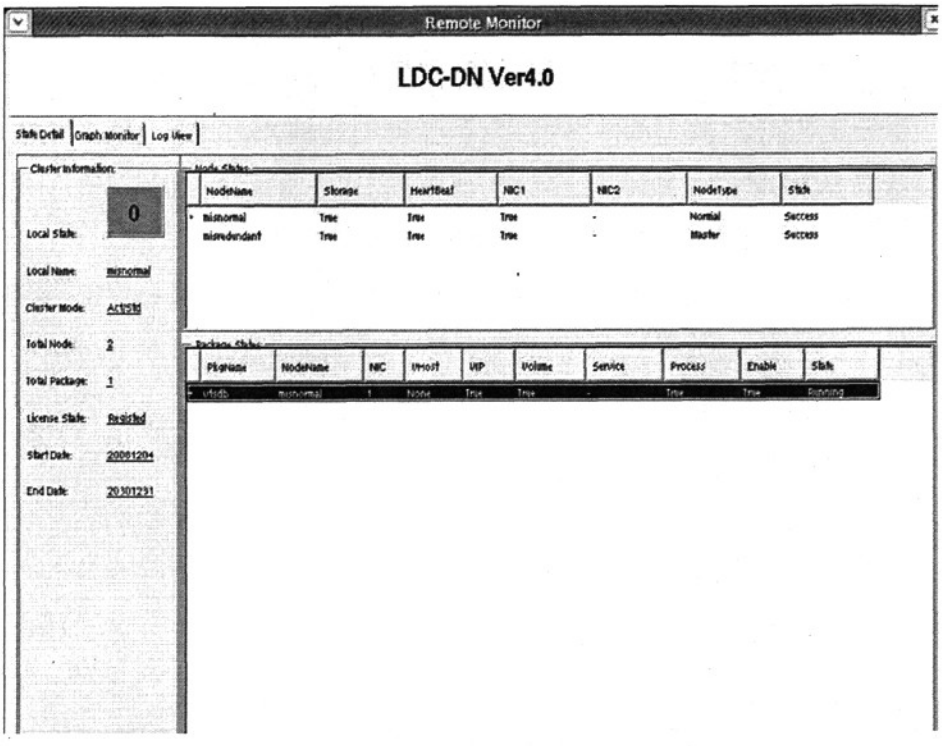

图 5.4 系统运行

Fig. 5.4 System operating

5.3 VTS 双机热备系统的测试

VTS 双机热备系统实现后要对其的功能及可靠性进行测试和分析。测试其功能时采用的是对系统在实际运行过程中可能发生的各种故障进行人为的模仿[50]，这种方法通常叫做故障注入法(Fault Injection)。测试双机热备系统是否能及时发现故障并进行自动切换，并对其可靠性进行分析。

为了测试双机热备系统的功能，人为的制造一些系统运行时经常发生的问题

来测试双机热备系统对各种故障是否能够进行准确的检测和及时的任务接管以及整个双机热备系统的应用服务恢复时间。如果对于某种故障，系统并未监测到，而系统此时又没有正常运行工作，则需要记录下人工发现故障并排除以及双机热备系统恢复正常的时间。诊断、任务接管和恢复过程都会影响系统的工作，其时间长短都将影响双机热备系统的可用性及可靠性指标。

在 VTS 双机热备系统中，故障的侦测有两种[51]：一种是服务器节点自身的侦测，服务器节点按规定的间隔对应用程序、系统资源等进行侦测，如果判断出故障则应按用户的选择切换到备份机工作或者仅发出相应警告。如果切换，则以备份机成功地接管工作机的应用服务为标准；另一种是服务器节点之间的相互侦测，一台服务器节点按一定的时间间隔通过心跳线发送心跳信息到另一台服务器节点，以收不到对方服务器节点的信息为系统故障的依据。

因此本文的测试将分为硬件和软件两个方面来进行。

1、硬件故障的测试

硬件故障可分为服务器故障、心跳线故障、阵列柜故障三个方面去进行测试：(1) 服务器故障。主机的故障如主板、电源、CPU、总线等故障均可导致系统中的一台服务器无法收到另一台服务器的心跳信息从而导致故障接管。测试时将主服务器正常关机和非正常关机，如非正常断电。测试能否切换到备用服务器和切换的时间；备机测试同主机类似，测试是将备份服务器正常关机和非正常关机，测试主服务器能否监测到故障和监测到该故障的时间。(2) 心跳线故障。拔掉一根心跳线，串口或者内外线。测试双机热备系统对心跳线故障的反应及反应时间；同时拔掉两个心跳线。(3) 阵列柜故障。拔掉 SAS 线，测试双机热备的反应；关掉阵列柜的电源，测试系统。

2、软件故障的测试

对于软件故障的可分为应用程序的故障、数据库故障、操作系统故障三个方面进行测试：(1) 应用程序故障。在 VTS 双机热备系统中，监控功能对相关系统服务的应用进程进行监控。当应用进程发生故障时，系统会根据应用进程的状态，执行相关的处理，必要时会进行切换。测试要求为：当监控进程本身失效后，系

统切换并接管相应的服务；当应用程序发生故障后，系统切换并重新启动该应用程序。所以对应用程序故障测试的方案为：①kill 掉监控进程；②kill 掉一个应用进程；③制造一个僵死的应用进程。测试系统应用的工作状态和主备服务器的状态。记录以上故障切换和恢复的时间。(2) 数据库故障。对于数据库的故障，设计了两个测试方案：①将数据库正常关闭；②将数据库进程非正常的 kill 掉。检验主备服务器是否正常切换和记录切换的时间。(3) 操作系统故障。如果操作系统发生故障可能会导致应用系统的严重故障，测试时杀死系统关键进程致使系统停机，检验主备服务器是否正常切换和记录切换的时间。

硬件故障测试结果如表 5.5：

表 5.5 硬件故障测试表

Tab. 5.5 Test table of hardware failure

测试类型	编号	测试内容	测试结果
服务器测试	1	主服务器正常关机	正常切换，服务正常接管；时间 19s
	2	主服务器非正常关机	正常切换，服务正常接管；时间 23s
	3	备用服务正常关机	检测到故障，时间 8s
	4	备用服务非正常关机	检测到故障，时间 10s
心跳线测试	5	拔掉串口的心跳线	检测到故障，时间 9s
	6	拔掉内网心跳线	检测到故障，时间 8s
阵列柜测试	7	拔主机与阵列柜的 SAS 线	正常切换，服务正常接管；时间 17s
	8	关闭阵列柜的电源	不切换，系统失效

软件故障测试结果如表 5.6：

表 5.6 软件故障测试表

Tab. 5.6 Test table of software failure

测试类型	编号	测试内容	测试结果
操作系统测试	9	终止操作关键进程	正常切换，服务正常接管；时间 21s
数据库测试	10	数据库正常关闭	正常切换，服务正常接管；时间 20s
	11	Kill 掉数据库相关进程	正常切换，服务正常接管；时间 22s
应用程序测试	12	Kill 掉监控进程	正常切换，服务正常接管；时间 23s
	13	Kill 掉 JAVA 进程	正常切换，服务正常接管；时间 19s
	14	Kill 掉 Tomcat 的进程	正常切换，服务正常接管；时间 22s
	15	造成应用进程僵死	正常切换，服务正常接管；时间 29s

通过表 5.5、表 5.6 的测试结果可以得出如下测试结论：对于我们注入的各种故障系统能够进行准确的诊断，并能够对各种故障完成正确的反应。系统的切换时间在我们能够接受的范围内，已基本上达到了海事局相关部门的用户需求和设计目标。

5.4 改进前后可靠性的比较

通过系统测试和 Gartner Group 以及国内外的相关资料数据统计，应用进程发生故障的概率大约占总故障的 8%左右[31][37][38]，另外，通过增加心跳检测的方式，故障侦测的效率也有了提高。这里为了便于数值计算和比较，设定系统的检测率提高了 10%。下面是改进前后系统可靠度的对比情况，如表 5.7：

表 5.7 改进前后可靠度数据表

Tab. 5.7 Reliability data sheet of before and after improvement

时间 t	θ=0.9 λ=0.0001	θ=0.8 λ=0.0001
单位 h	R(t)	R(t)
0	1	1
500	0.9968	0.9963
1000	0.9906	0.9881
2000	0.9845	0.9789
6000	0.9602	0.9432
9000	0.9407	0.9211
12000	0.9214	0.9016
15000	0.9107	0.8925
20000	0.9039	0.8807

在 MATLAB 的命令窗口中通过 plot 命令来绘制改进后 λ=0.0001 θ=0.9（实线）和改进前 λ=0.0001 θ=0.8（虚线）可靠度的曲线，如图 5.5。

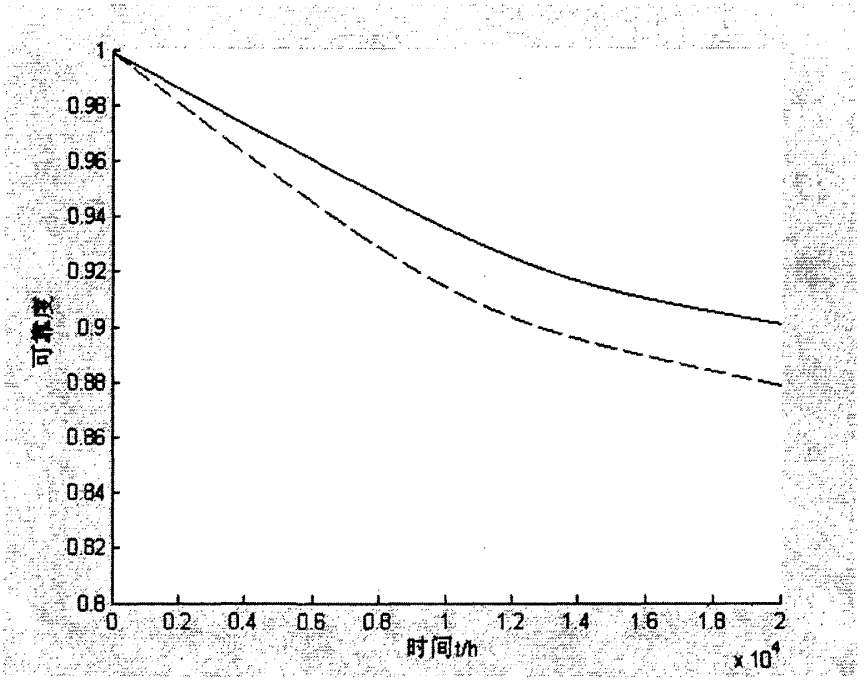

图 5.5 可靠度曲线图

Fig. 5.5 The figure of reliability curve

通过比较改进前后系统可靠度的数值和曲线，可以看出，当 VTS 双机热备系统的运行时间达到 $2*10^4$ 小时的时候，可靠度比改进前高出 0.0232，即 2 个百分点之多；并且由曲线图可以判断，随着时间的继续增长，可靠度下降的趋势比改进前也有所减缓。可见，当检测率提高时，双机热备系统的可靠性也会随着提高。

目前，此系统已经在海事局成功部署运行，用户反应良好，为 VTS 系统的稳定运行和数据安全等方面提供了保障。近期国内的部分海事局相关部门都采用的是本文的双机热备系统作为其服务器平台，可见，双机热备在海事局 VTS 系统中有着非常广泛的应用前景。

第 6 章 总结与展望

6.1 工作总结

本文所实现的双机热备系统目前已经成功的部署到海事局的现场环境中，并已稳定运行。它是海事局 VTS 系统的后台数据服务器，有着非常重要的作用和地位。本文在双机热备系统研究和应用中做了以下的工作：

1、双机热备相关理论的介绍。阐述了双机热备的相关概念，重点介绍了 RAID 磁盘阵列的相关技术，本文正是采用目前所流行的 RAID5 磁盘阵列来实现双机热备系统。在分析双机热备的作用和实现方式基础上，阐述了双机热备的关键技术。

2、VTS 双机热备系统的分析设计。双机热备系统是 VTS 系统实现高可用性的核心，本文给出了双机热备系统的设计方案，包括系统设计原则、工作模式，系统结构层次设计以及工作流程设计等，并根据设计的要求，分析了采用的双机软件模块功能。

3、VTS 双机热备系统的可靠性研究。采用 Markov 预测法对 VTS 双机热备系统建立了相应的可靠度模型，并根据可靠性的概念定义了两个参数，通过分析故障率和检测率这两个参数对系统可靠性的影响，为提高 VTS 双机热备系统可靠性的改进措施提供了的理论来源。在分析了影响系统可靠性因素的基础上，通过加入监控应用进程状态的功能和增加一条心跳链路来改进 VTS 双机热备系统的检测率，进而提高系统的可靠性。

4、VTS 双机热备系统的应用与测试。较为详细的给出了 VTS 双机热备系统的配置和实现过程，部署到海事局现场环境中，并给出了相关的脚本和配置文件。最后用故障注入法对系统功能进行了相关测试，并通过比较验证系统可靠性的提高。

6.2 下一步工作

本文通过对双机热备系统的相关理论进行了研究和分析，证明双机热备是关键性事务处理中广泛应用的一种技术，它实现的是系统级的冗余，通过快速的故障诊断与自动接管，以较低的成本获得较高的系统可靠性。双机热备系统的实现

涉及到相当多的技术问题。但由于时间和资源有限，本文仍然存在一定的不足，需要在以下两个方面深入研究：

第一，向多节点集群的“平滑扩展”。VTS 双机热备系统是以双机模型为基础的，它的优点是简单可靠；但是对于更大限度的提高系统可用性、面向大规模的集群应用来说，还有很多不足。

第二，可靠性研究的深化。由本系统构建的 Markov 模型设定的条件较为理想，相关的数据也需要进一步的测试和比较。所以在描述系统的 Markov 过程时可以设置更多的参数将其状态细化，从而更加准确的描述双机热备系统的状态转移。另外，如果对多点集群可靠性建模分析的话，可以采用 Markov 模型与动态故障树模型相结合的方法来研究，但分析和求解过程会更加复杂。

总之，随着各行各业对于服务器性能和数据安全的需求不断增强，相信本文能够起到一定的参考作用，也希望有更多的新技术融入到这一领域，实现更加完美的解决方案。

参 考 文 献

[1] 唐强荣. VTS系统结构模型. 广州航海高等专科学校学报, 2001, 2:26-28.

[2] WenXiaofei, SongZongbo. Performance evaluation of high performance computer cluster. Journal of Wuhan Automotive Polytechnic University. 2005, 1(4):24-30.

[3] 郑纬民. 集群系统的现状与挑战. 计算机教育, 2004(6):23-24.

[4] 赵殿奎. 基于LVS 负载调度器的双机热备份研究与实现:(硕士学位论文). 大连:大连理工大学, 2006.

[5] http://www.statsoft.com/textbook/stcluan.html

[6] Hennessy J, Patterson D. Computer Arhcitecture:A Quantitative Approach. 北京:机械工业出版社, 2002.

[7] Marc F, Tom S, Jeffrey H. Times guide to security and data integrity. 李明之等译. 北京:机械工业出版社, 2005.

[8] 程明华, 姚平. 动态故障树分析方法在容错计算机系统中的应用. 中国航天学会控制与应用第八届学术年会. 北京航空航天大学, 1998:63-68.

[9] 蒋乐天, 徐国治, 应忍冬. 系统可靠性和可用性分析技术. 电信技术, 2002(4):121一123.

[10] 高文, 祝明发. 基于生灭过程的机群系统高可用性分析与设计. 微电子学与计算机, 2001(4):48-49.

[11] 蒋乐天, 徐国治, 应忍冬. 随机Petri网在系统可用性分析中的应用. 系统仿真学报, 2002, 14(6):196-198.

[12] 姚颖熹. LINUX下双机数据热备份的设计与实现:(硕士学位论文). 成都:电子科技大学, 2001.

[13] http://blog.csdn.net/taige5555/archive/2008/11/19/3334773.aspx

[14] http://baike.baidu.com/view/7102.htm

[15] Robertson B. A Highly-affordable and High availability. Linux Magazine, 2003, 11:1-13.

[16] 陈华英.磁盘阵列 RAID 可靠性分析.电子科技大学学报,2006.6,35(3).

[17] Amnon Barak, Sean Borgstrom, Baruch Awerbuch. An Opportunity Cost Approach for Job Assignment in a Scalable Computing Cluster. IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 11. NO. 7, July 2000.

[18] Vinod C. Reliability and Safety Analysis of Fault Tolerant and Fail Safe

Node for using in a Railway Signal Systems. Reliability Engineering and System safety, 2001, 57:177-183.

[19] http://www.redhat.com/solutions

[20] 戎强. 双机热备份应用模式数据库解决方案. 中国金融电脑, 1998, 01.

[21] 赵昆. 纯软件方式容错系统的实现. 计算机世界, 2005.

[22] 侯新峰. 基于LINUX的心跳基础平台设计与实现:(工程硕士学位论文). 长沙:国防科技大学, 2004.

[23] Wang Y M, Huang Y. Check pointing and its applications for Fault-Tolerant Computing System. 2001:22-31.

[24] 申志冰, 罗宇. 利用 Heartbeat 实现 Linux 上的双机热备份系统. 长沙:国防科技大学, 2002.

[25] Chen C H, Ting Y, Lu W B. Recovery mechanism design for hot standby computer system. IEEE Computer Society. 2003, 3(3):3027-3031.

[26] 田灼. 双机容错热备份系统的研究与实现:(硕士学位论文). 哈尔滨:哈尔滨理工大学, 2003.

[27] Matthew L, Brent N, David E. The distributed monitoring system:design, implementation, and experience[J], Parallel Computing. 2000.

[28] http://lcic.org/load_balancing.html

[29] 李大夜. 基于 linux 的集群和心跳设计:(硕士学位论文). 哈尔滨:哈尔滨工业大学, 2006

[30] http://linux.chinaunix.net/bbs/thread-903138-1-1.html

[31] 孙青, 庄奕琪等. 电子元器件可靠性工程. 北京:电子工业出版社, 2002, 10:48-54.

[32] 金士尧, 胡华平. 具有容错结构的高可用计算机双系统研究. 中国科学工程, 2000, 12(3):46-48.

[33] 崔锦龙, 邓姝杰. 基于Markov数学模型的降水预测及其利用. 资源与开发市场, 2008, 24(2):87-89.

[34] 白立军, 张银福. 一种双机热备机群的可信性建模分析. 长沙:国防科技大学, 2007.

[35] 胡华平, 肖晓强. 重构双机系统的可靠性分析. 航天控制, 2007, 15(3):16-18.

[36] 闫剑平, 汪希时. 两种方式双机热备结构的可靠性和安全性分析. 铁道学报, 2000.

[37] Asadi M, Bayramoglu I. The mean residual life function of the structure at the system level[J]. IEEE Transactions on Reliability(S0018-9529), 2006, 55(2):314-318.

[38] Micliele F, Cecilia M. TMR voting in the presence of crosstalk faults at the voter

inputs[J]. IEEE Transactions on Reliability(S0018-9529), 2004, 53(3):342-348

[39] 刘新宇, 高文, 孙凝晖. 双机热备份集群的可信性建模分析与比较. 小型微型计算机系统, 2004, 25(4):747-751.

[40] 苏金明, 刘宏, 刘波. MATLAB 高级编程. 北京:电子工业出版, 2006.

[41] 陈宝平. 话单采集双机备份的研究与实现:(硕士学位论文). 大连:大连海事大学, 2005.

[42] http://www.linuxjournal.com/article/3247

[43] Lars R. A Network on a Chip International Conference on Parallel and Distributed Processing Techniques and Applications. 2002.

[44] http://forum.ubuntu.org.cn/viewtopic.php?f=139&t=170407

[45] Arie Keren, Amnon Barak. Opportunity Cost Algorithms for Reduction of I/O Interprocess Communication Overhead in a Computing Cluster. IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 14. NO. 1, January 2003.

[46] 李双庆. WEB服务器集群技术研究:(博士学位论文). 重庆:重庆大学, 2003.

[47] landercluster 技术白皮书. 上海联鼎技术有限公司. 2008.

[48] Christine Morin, Pascal Gallard. Towards an efficient single system image cluster operating system. Future Generation Computer Systems, 2004(20):505-521.

[49] URL:http://repositories.cdlib.org/cgi/viewcontent.cgi

[50] 毛少杰. 基于仿真的 C^3I 系统测试评估技术研究. 系统仿真学报, 2003, 15(6):26-27.

[51] 李智, 王向君. 微机电系统测试技术及方法. 光学精密工程, 2003, 11(1):37-44.

致　谢

在本文的 VTS 双机热备系统的研究与实现中，一直得到了蒋剑平老师的指导和关心，前后经历了一年多的时间，他渊博的学识以及深厚的专业素养不但帮助我拓宽了研究问题的思路，而且还提高了自己分析问题的能力。从他那里，我不仅得到了许多技术上的具体指导，也学到了作为一个研究人员所应具备的严谨踏实的学习和工作作风。他诲人不倦的仁厚长者风范与严谨的治学态度给我留下了深刻的印象，这必将伴随我的一生，时刻勉励自己，终身受益。

在这里我还要感谢给予我极大支持和鼓励的实验室同学：李栋，董恒，许榕夏以及其它实验室的同学：肖智博、魏善岭等。我的课题能够顺利完成与他们是分不开的。

同时，本文的完成还离不开海事局工作人员的大力支持和配合，在这里对他们也表示衷心的感谢。

最后，向评审本文的各为专家致意！

研究生履历

姓　　名	邢传星
性　　别	男
出生日期	1983 年 05 月 10 日
获学士学位专业及门类	理学
获学士学位单位	辽宁师范大学
获硕士学位专业及门类	工学
获硕士学位单位	大连海事大学
通信地址	辽宁省大连市凌海路 1 号
邮政编码	116026
电子邮箱	xingchuanxing@126.com