基金颁发部门：信息产业部 242 基金；基金申请人：刘晓洁　颁发年月：2006.2

一种基于 Internet 的容灾系统关键技术

刘晓洁

（四川大学 计算机学院，四川 成都 610065）

（liuxiaojie8@126.com）

摘　要：本文提出并实现了一个基于非专线 Internet 的远程容灾系统。通过研发的本地数据监控和异地重放、差错控制、服务切换等关键技术，解决了远程容灾中本地数据带宽与备份线路带宽不匹配这一传统技术困难。系统测试结果显示该方法适用于 Internet 环境下的容灾解决方案。

关键词：容灾抗毁；数据监控；差错控制；服务切换；

中图分类号：TP309　　**文献标识码**：B

Key Technologies for a Disaster Tolerant System Based on Internet

Liu Xiaojie

Computer school of Sichuan Univ., Chengdu, Sichuan, 610065, China

Abstract: A remote disaster tolerant system based on Internet is proposed and realized. Building on the self-developing key technologies for local data monitoring and remote data re-writing, error control and service switch, the traditional problem that their bandwidth do not match between local and remote network has been solved. The experimental result shows that the proposed method is available for building disaster tolerant system on Internet.

Key words: disaster tolerance ; data monitoring; error control; service switch

1 引言

随着信息系统日益占据着企业竞争和国家综合国力竞争的主体地位，容灾已成为信息安全领域重要的研究方向。一个真正意义上的容灾系统应提供异地容灾功能，目前国内外绝大多数异地容灾为了减少对原系统的影响都是基于专线或光纤通信等特殊设备，价格昂贵。基于非专线的且对原系统影响小的异地容灾系统的研制还处于起步阶段。

针对目前由于专线价格因素，导致国内中小企业拥有远程容灾系统少之又少的现状，通过对远距离容灾关键技术的研究，本文设计并实现了一个低成本的、对原系统影响小的可基于 Internet 的远程容灾系统 DRC。

2　DRC 系统结构

图 1 是整个 DRC 系统的体系架构图。它由位于本地的生产中心和位于异地的容灾中心组成。其中生产中心由本地服务器群和本地“灾备网关”以及相应系统软件组成；容灾中心由

基金项目：信息产业部 242 基金

作者简介：刘晓洁（1965-），女，副教授，主要研究方向：网络安全技术及应用。

异地服务器群和异地“灾备网关”以及相应系统软件组成。本地（异地）服务器与本地（异地）“灾备网关”之间，通过内部高速网络连接，生产中心和容灾中心之间通过 Internet 连接。

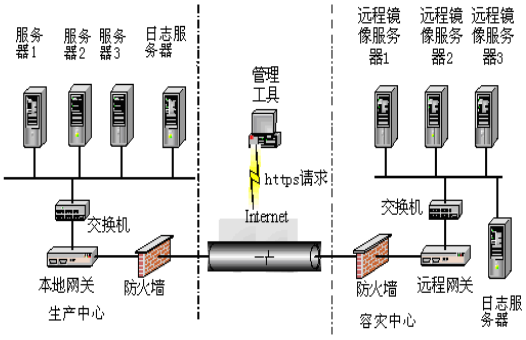

图 1 容灾系统 DRC 拓扑结构

正常情况下，由本地服务器提供服务，本地服务器数据的改变通过数据监控技术被实时送达本地灾备网关，该网关通过海量数据高速缓存技术，采用异步通讯模块和独特的差错控制技术将数据通过 Internet 发送到远程灾备网关，远程灾备网关采用异地重放技术在异地服务器相应的分区上按次序重写写操作，从而达到实时数据备份的目的。当本地服务器发生灾难时，远程服务器通过服务切换技术，自动切换对外提供服务，使外界觉察不到服务的中断。

3 DRC 关键技术

DRC 系统通过本地数据监控、异地重放、差错控制、服务切换等关键技术，在不影响本地数据中心服务器效率的情况下，实现本地数据中心数据在异地的实时重构以及服务的切换，解决了本地数据带宽与备份线路带宽之间的突出矛盾这一技术问题，从而可直接架构于非专线的 Internet 上。

3.1 数据监控及异地重放技术

采用数据监控技术对生产中心数据的变化进行精确的监控，全面记录本地数据中心数据变化的详细过程，确保在数据镜像时滴水不漏。这里的关键技术是数据监控点的设置。目前的应用大都将点设在文件系统层，而在 DRC 系统中通过将监控点设在块设备层之上，实现了数据监控对上层应用透明，与文件系统无关，与底层物理磁盘类型无关。特别是对于有些数据库系统，如 Oracel Mysql, 他们有自己的读写机制，其读写操作并不通过文件系统而直接读写磁盘逻辑卷，常规的文件系统层监控点就束手无策了。

DRC 数据监控的主要实现思想是：针对本地服务器上需备份的块设备，数据监控模块 DMON 在设备驱动层截获磁盘写操作 （将监控点设定于此，可起到对上层文件系统及下层具体物理介质透明的效果，从而使得系统达到一定的跨平台的功能）， 并将相关信息如写数据及柱面、扇区、磁道等封装成重放记录 gw_bh，发送到本地灾备网关上缓存。由于系统仅缓存了本地服务器的写操作数据，即仅备份了本地和异地服务器之间的差异数据，从而通过这种差异备份，减少了网络传输及备份数据量。降低了网络带宽要求。

同时通过二级转发数据海量缓存技术，本地服务器仅需通过高速网络将差异数据送入本地灾备网关缓存，余下的远程备份工作由该网关完成，当缓存足够大时，即使在 Internet 这种不太稳定的复杂网络环境中，也能减少对本地服务器的效率影响。

异地重放技术的关键在于远程灾备网关重写差异数据到远程服务器时，应保证和本地服务器的写顺序严格一致，针对每个重放记录对远程服务器完成一次写操作。

3.2 差错控制技术

在异地重放过程中，针对 Internet 的不可靠性，需要进行严格的数据差错控制，以确保本地、异地数据的一致性。容灾系统中检验本、异地数据是否一致的常用的方法为通过比较变化数据摘要值和镜像对象数据状态摘要值来实现。如果发现摘要值不一致，则表明出现数据不一致性，需要请求重传、进行数据同步。

检测备份数据正确性的传统手段是采用事后比较的方法，但此法不能及时发现数据备份过程中发生的差错。我们希望能够在每一条备份记录成功保存到远程端时，都能保证此时远程端的数据与某一时刻本地端的数据完全一致。为了达到此目的，本系统采用了累计散列值算法。

所谓累计散列简单的讲就是散列的散列，它将上次的散列结果作为部分输入数据参与本次的比较，其思想是避免对已经确认一致的数据进行重复计算。比如，在 t1 时刻，本地数据与远程数据的散列值分别是 EL1 和 ER1，且 EL1==ER1。在 t2 时刻，本地成功传输若干数据到远程，此时，重新计算两端的累计散列值有二种选择：①分别计算 H(ds1+Δdata)和 H(ds1'+Δdata')，其中 H 代表散列算法，如 MD5、SHA-1 等，ds1 和 ds1'代表 t1 时刻本地与远程的数据快照，Δdata 和 Δdata'代表 t2 时刻本地与远程的数据变化；②分别计算 H(EL1+Δdata)和 H(ER1+Δdata')。这两种方法的效果一样，但第 2 种方法明显能够节约时间，提高效率。

3.3 服务切换

本地服务器出现故障时，服务切换技术能够将服务切换到远程服务器，由它对外提供不间断的服务。实施服务切换具体步骤为：当通过心跳技术，监测到本地服务器出现故障时（即失效时），启动容灾中心服务器的服务，修改目标系统的访问路径。DRC 系统采用基于 IP 重定向的服务迁移技术，实现了应用服务由本地生产中心自动切换到异地容灾中心的功能。

服务切换中的关键技术是失效检测算法。传统的失效检测算法一般采用基于 PUSH 模型、或 PULL 模型的失效检测算法。当检测器在规定的时间 deadtime 内未收到 aliveness（存活）信息时，则判断待检测端失效，服务切换。但实际网络传输中存在数据包延时、丢包等导致的伪失效情况，传统的失效检测算法没能考虑到这点，从而可能由于失效误判，导致大的服务切换开销。

鉴于此，在 DRC 系统中, 采用了基于 PUSH 模型和 PULL 模型相结合的复合失效检测算法，引入怀疑（SUSPECT）状态。在第一阶段中，待检测端使用推模型，因此它发送 aliveness（存活）信息给检测器。若在规定时间 deadtime 内收到 aliveness 信息，则认为待检测端处于正常（UP）状态。 在一段延迟后，检测转为第二阶段，在这个阶段里，第一阶段中没有发送存活信息的超时待检测端并不立即判为失效，而是判为怀疑（SUSPECT）状态。这时对该待检测端使用拉模型，检测器发送 aliveness 信息给待检测端，并且期望从待检测端上收到 aliveness 信息。如果待检测端没有在一定时间限制内发送这种信息，这时判断其已失效（DOWN 状态）。具体算法描述如下：

FD()
{
PUSH:
receive message m_k at time t from s_i($s_i \in S$);
g_i.timer= 0; //reset timer
g_i.state = UP;
$g_i.E_{(k+1)}$=System Estimate Value;
PULL:
For($g_i \in G$)

{

g_i.timer++;

If((g_i.timer > gi.$E_{(k+1)}$)&&(g_i.state== UP))

{

gi.state =SUSPECT

send message to s_i;

g_i.$E_{(k+1)}$+=RRT of message;

}

Else

if((g_i.timer>g_i.$E_{(k+1)}$)&&(g_i.state=SUSPECT))

g_i.state =DOWN

}

}

其中 FD 表示失效检测器模块，S 表示本地服务器的待检测服务列表，RRT 表示检测端与接受到待检测端的 aliveness 信息之间的延时。g_i 表示 s_i（$s_i \in S$）的状态信息，g_i 的结构为：

struct g_i {

s_i; //对应的本地服务器产生的服务

timer; //此检测器的定时器

state; //本地服务器的状态

$E_{(k+1)}$; //预计下次检测的时间

}

同时当网络状态长时间不稳定，网络丢包率高时，可动态调整延长 deadtime 值，减少失效误判，从而更好的适应现实 Internet 网络环境。

3.4 系统性能测试和分析

在本地/异地服务器与本地/异地灾备网关为 100M bps 相连，两个灾备网关通过 512K ADSL 及 100M 专线相连的网络环境中，对异地容灾系统 DRC 的数据存储时间性能进行了测试。测试环境：本地/远程服务器均为 1.7GHZ、256M、Linux 2.4.32 内核，本地/远程网关均为 2.8GHZ、512M、Linux 2.4.32 内核。

在上述测试环境下，分别采用大小为 100MB、500MB、1GB、1.5GB、2GB 的文件对本地服务器的数据存储时间进行测试。测试结果见图 2。

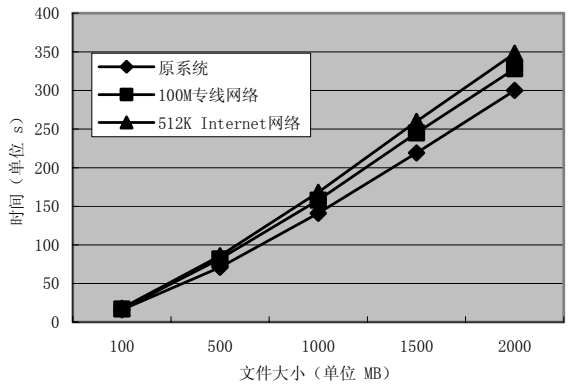

图 2 系统性能测试图

从图 2 中可以看出，DRC 系统比原系统（未运行 DRC 时的系统）的数据存储时间增多不到 16%，同时比对本、异地通过 100M 专线连接和 512K ADSL 非专线连接网络环境中，数据存储时间的增加不到 6%，且随着存储数据量的增大，时间增长率保持稳定。可见，本容灾系统的异地备份对原系统的性能影响小，而且完全可适用于 Internet 非专线网络环境中。

4. 结束语

目前市场上流行的容灾系统大多为国外大公司所垄断，研究容灾系统的核心技术，对建设具有我国自主知识产权的容灾系统具有积极的意义。采用本文研发实现的数据监控、异地重放、差错控制、服务切换等关键技术，可使得容灾系统能基于非专线的 Internet 网络，降低了常规异地容灾对网络带宽和稳定性的要求，同时相比国外同类产品，大大降低了成本。

参考文献：

[1] 潘爱民，阳振坤. 关于灾难恢复计划的研究[J]. 网络安全技术与应用, 2005, 2003(2): 23-26.

[2] 王琨，袁峰，周利华. 灾难恢复系统模型研究[J]. 网络安全技术与应用, 2006, 2006(3): 10-13.

[3] 李强，张艳，李舟军. 用于灾难恢复的远程备份系统的模型与算法[J], 计算机工程与科学, 2005, 27(5): 68-72.

[4] Fallara, P. Disaster recovery planning[J] . IEEE Potentials, 2004 , 22(5): 42-44.

[5] CORBET J,RUBINI A,KROAH-HARTMAN G. Linux 设备驱动程序[M]，魏永明,耿岳,钟书毅译. 北京：中国电力出版社,2006.

[6] 贾云洁，欧阳旦，傅永刚. 远程备份监控系统的设计与实现［J］. 微计算机信息，2006，8-3：177-179

作者简介：

刘晓洁(1965-)，女，江苏南京人，副教授，研究方向：网络安全技术及应用

Biography: Liu xiaojie, female, Nanjing Jiangshu, Sichuan University, vice professor, major in network security

本文创新点：

① 在本地数据监控模块，巧妙地将数据监控点设置在块设备层之上，克服了目前的应用大都将点设在文件系统层的缺点，实现了数据监控对上层应用透明，与文件系统无关，与底层物理磁盘类型无关。

② 通过仅备份差异数据的高速缓存技术解决了本地带宽和 Internet 带宽的不匹配问题。

③ 通过累计散列值算法克服了传统的采用事后比较检测备份数据正确性的方法，可及时发现数据备份过程中发生的差错。克服了 Internet 的不稳定性。

④ 采用了基于 PUSH 模型和 PULL 模型相结合的复合失效检测算法。引入了怀疑（SUSPECT）状态，降低了传统 PUSH 或 PULL 模型对失效状态的误判，克服了误判造成的较大服务切换开销。