

# 医院信息系统中灾备系统的设计与实现

刘晓辉

(广州军区广州总医院信息科 广东 广州 510010)

**摘要:** 医院信息系统是现代医院运营的数据基础平台, 保持系统的可用性和业务的连续性变得越来越重要, 建立灾备系统能使医院在遇到意外灾难时, 保障业务的连续性, 最大程度地降低损失。灾备系统可以采用多种技术方案, 本文采用了硬件灾备同软件灾备相结合的方式提供对系统的保护, 保证系统高可用性和对灾难的抵御能力。该方案在一所大型医院的 HIS 系统得到应用, 多次有效的避免了各种原因引起的宕机时数据的丢失。

**关键词:** 医院信息系统 灾备系统

Design and implementation of disaster recovery system in HIS

**Abstract:** This article protects the HIS by using software addition hardware disaster recovery system. This solution has already been put into a production system in a large hospital information system. It's proved to be a good solution in many areas.

**Key words:** hospital information system Disaster recovery system

## 1. 引言

随着医院信息化建设的不断深入, 范围的不扩大, 各种业务系统所产生的核心数据逐渐增多。医院的正常运作已离不开这些系统和数据的支撑。因此, 各种意外情况、硬件的损坏以及人为误操作等等所带来的数据丢失和业务停顿都可能给医院造成无法挽回的经济损失。

对于医院信息系统(HIS)的维护人员来说, 保持系统的可用性和业务的连续性变得越来越重要, 医院通常 24 小时开诊, 这就要求 HIS 系统 7x24 小时完全在线运行。不仅要考虑整个系统的备份问题, 同时还要保证在生产系统出现故障时能保持业务的连续性和数据的完整性。因此, 建立以备份和灾难恢复为目的系统, 可以为整个医院信息系统提供安全保障, 一旦生产系统出现故障, 可以利用灾备系统进行恢复, 将故障对业务的影响到减到最低, 从而实现最小的业务停顿时间。

## 2. 灾备系统现状

银行、电信等大型企业所拥有的系统都有较完善的灾备系统, 相比较而言, 由于医院受经费、技术条件等限制, 医院信息系统(HIS)中通常只有日常的备份系统在运作, 灾备系统还只是一个概念。但随着 HIS 系统在医院的地位越来越重要, 医院运营的数据一旦受到破会后如果无法得到及时恢复, 将会对医院产生无法挽回的损失。因此, 建立灾备系统, 防患于未然, 将为 HIS 系统的长久稳定的运行提供有力的安全保证。

某院信息系统（HIS 系统）1999 年正式投入使用，随着系统规模的不断扩大，目前已发展成为拥有四个大系统，一千余台客户端的中等规模的 HIS 系统。

## 2. 1 医院信息系统硬件和数据库系统

该院现有主信息系统由一台 IBM X445 服务器加上 MSA1000 阵列构成，医院的主要业务系统一门诊、住院、收费等应用都运行在该服务器上。该服务器上安装的是 Windows2000 操作系统，数据库版本为 Oracle8174，现有数据量约为 34G。

目前应用一台 HP LH6000 服务器做为备用服务器，当主服务器 IBM445 出现硬件故障的时候能顶替上线。

## 2. 2 现有系统备份状况简介

数据库运行在归档模式下，目前的备份方式为全库冷备，备份策略是每周六停机一次，将所有的数据库文件拷贝到本机另一个空闲盘上，然后启动数据库，同时将该备份再次拷贝到远程的备份服务器和备份测试机器上，确保多个备份同时存在，并将备份测试机上的备份打开查看是否备份成功。

## 2. 3 不足之处

- 从目前的备份方式来看，整个系统只有备份系统，容灾系统还没有建立；
- 备份系统为人工手动冷备份，有人为误差的可能性，管理也较复杂；
- 备份间隔时间长，从而导致恢复时间长，难以适应目前业务系统需要；
- 无法实现增量备份，导致每次备份时间过长。

## 3. 灾备方案设计原则

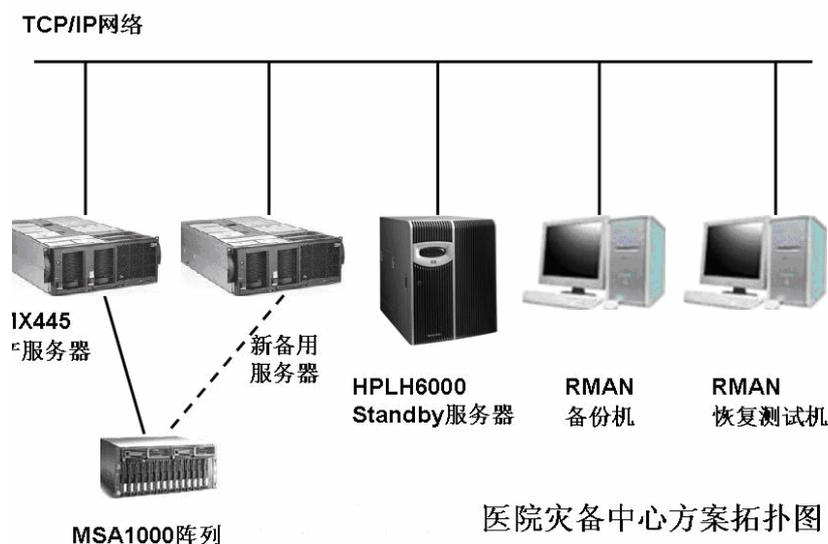
根据医院现有备份系统的现状和不足，我们遵循以下原则设计了灾备方案：

- 增加容灾系统，使灾备中心具有相对完善的数据保护能力；
- 尽可能实现备份和容灾的自动操作，减少人工可能造成的失误；
- 能对已完成的备份进行检查；
- 在系统出现故障时，Standby 系统能尽快顶替上线把停机时间和损失降到最低；
- 充分利用现有空闲的一台 HP LH6000 服务器，尽量减少对现有系统的大规模改变，保证企业投资回报率。

此外，在系统建设期间，应保证所有业务的正常运行；在系统使用期间，采取一定保护措施，保证在意外发生的情况下将系统恢复到原有状态。

## 4. 灾备系统建设方案

新的灾备系统在充分利用原有设备的基础上，增加部分新设备，把备份系统由原来的冷备方式改为 RMAN 方式，同时建立具有灾难恢复功能的 STANDBY 系统，整个 HIS 系统灾备方案拓扑图如下。



#### 4.1 备份系统方案

RMAN (Recovery Manager) 是 ORACLE 提供的一个重要备份工具，属于在线热备份系统，因此，用 RMAN 进行系统热备份时能将备份时对系统的影响减到最少。

RMAN 的备份实施方案包括配置 CATALOG 数据库，制定 RMAN 的多级备份，以及写 RMAN 备份脚本。

##### A. 配置 CATALOG 数据库

因为 RMAN 能自动维护备份和恢复所需要的各种信息，所以 RMAN 必须把这些信息以某种形式保存。RMAN 支持两种形式保存这些信息：数据库的控制文件或创建一个单独的数据库来保存 RMAN 的信息。当选择把 RMAN 的信息存储在控制文件时，控制文件的丢失时将导致备份将不能进行恢复。所以若采用 RMAN 做备份，一定采用 RMAN CATALOG 数据库来单独存放备份信息。这个单独的数据库（称为 CATALOG 数据库）只需要很小的空间，在本方案拓扑图中可以看到 CATALOG 数据库存放在单独的一台 PC 机器（RMAN 备份机）上。

因为 CATALOG 数据库包含了所有的备份信息，所以该数据库本身也是需要通过某种方法进行备份，但因为该数据库很小（一年内可能才增加十几二十兆），所以既可以对它进行冷备份，也可以进行逻辑的输出（EXPORT）。CATALOG 数据库具体配置如下：

- 1) 用 dbassist 创建数据库。

- 2) 在该数据库创建 RMAN 数据库用户:

```
create user rman identified by rman default tablespace ts_rman
temporary tablespace temp;
grant connect ,resource , RECOVERY_CATALOG_OWNER to rman.
```

- 3) 连接到目标数据库和 CATALOG 数据库

```
rman target system/manager@target_tnsname catalog rman/rman@catalog_tnsname
```

- 4) 创建 CATALOG 用户的表:

```
rman>create catalog
```

- 5) 登记目标数据库:

```
rman>register database
```

这样就可以利用该 RMAN 数据库来备份目标数据库了。

#### B. 制定多级备份策略

- 1) 在 RMAN 备份 PC 机上建立 Catalog 数据库, 用来储存脚本和管理备份, 并将该 PC 机上的分区映射到 IBMX445 服务器上;
- 2) 用 Catalog 连接到生产库上, 将数据库备份到 PC 机上映射到 IBMX445 本地的磁盘。每次备份完成后可将备份拷贝到其他存储设备做一个备份;
- 3) 每次备份完成后将备份恢复到恢复测试 PC 机上。
- 4) 由于 RMAN 可以实现多级增量备份, 这样在制定具体的备份策略时可以更加灵活:

每一个星期天做一个数据库的全备份; 每星期三做一级备份; 每天做一个二级备份; 当需要时 (如三天归档文件系统就要接近满了) 备份归档文件。

数据库全备份脚本:

```
run {
allocate channel 'dev1' type disk ;
allocate channel 'dev2' type disk ;
backup full tag 'dbfull' format '/o9i/testrman/full%u_%p_%c'
database ;
sql 'alter system archive log current' ;
backup filesperset 3 archivelog all delete input;
release channel dev1;
release channel dev2;
```

```
}
```

#### 一级备份脚本

```
run {  
  allocate channel 'dev1' type disk ;  
  allocate channel 'dev2' type disk ;  
  backup incremental level 1 tag 'dbL1' format  
  '/o9i/testrman/basic%u_%p_%c' ddatabase skip readonly;  
  sql 'alter system archive log current' ;  
  backup filesperset 3 archivelog all delete input;  
  release channel dev1;  
  release channel dev2;  
}
```

#### 二级备份脚本

```
run {  
  allocate channel 'dev1' type disk ;  
  allocate channel 'dev2' type disk ;  
  backup incremental level 2 tag 'dbL2' format  
  '/o9i/testrman/basic%u_%p_%c' ddatabase skip readonly;  
  sql 'alter system archive log current' ;  
  backup filesperset 3 archivelog all delete input;  
  release channel dev1;  
  release channel dev2;  
}
```

#### 归档文件备份脚本

```
run {  
  allocate channel 'dev1' type disk ;  
  allocate channel 'dev2' type disk ;  
  backup filesperset 3 archivelog all delete input;  
  release channel dev1;  
  release channel dev2;  
}
```

根据上述备份策略，每天所需要备份的数据量只有一天的改变量。而做恢复时最多只要恢复当周的一个零级备份+1 个一级备份+3 个二级备份+当天的归档文件。如果不能接受这样的恢复时间，

就只能减少零级备份之间的时间间隔（如可以每个三天做一个零级备份，每天做二级增量备份，这样恢复时最多只需要三天的数据量）。

#### 4. 2 容灾方案

STANDBY 数据库是 ORACLE 数据库提供的一种数据库容灾方案。Standby 数据库通过不断应用主数据库产生出的归档日志文件来恢复到最新状态；主数据库处于归档模式，持续送出归档日志给 Standby 数据库，而 Standby 数据库则处于恢复模式，持续应用主数据库的归档日志。这种机制保证了两个数据库之间数据的一致性。因此，在主数据库出现故障时可以激活 Standby 数据库从而取代主数据库对外提供服务，也可以通过导出 Standby 数据库的数据对主数据库进行恢复。从而保证了系统的容灾性。本系统在容灾方案上采用在独立服务器上安装 STANDBY 数据库方式来实现软容灾与硬容灾相结合的方式，最大限度的降低了故障恢复时间。

STANDBY 数据库的容灾能力体现在以下几个方面。

在主数据库下线的情况下，可随时将 Standby 数据库恢复到最新的状态，然后打开顶替使用。整个过程只需要几分钟(具体的时间要看有多少归档日志需要恢复)；

1) Standby 数据库可以 Readonly 方式打开，运行一些报表查询、统计，或其他大量的、对实时数据要求不高的查询。但是 Standby 数据库一旦以 Readonly 方式打开，就不能马上起到容灾效果，在数据库宕机的时候不能及时顶替上线；

2) 可人工控制 Standby 数据库与主数据库的时间差距，如果在主数据库中错误地删除了一些数据，可以将 Standby 数据库以 READONLY 方式打开，然后将需要的数据 exp 出来。

#### 5. 小结

建立灾备中心的最终目的是为了在各种故障、灾难发生的时候，都能让业务保持 7X24 的不间断连续性。在数据库某些数据文件发生丢失或损坏的情况下，利用备份快速准确地恢复，减少数据丢失；在操作系统甚至服务器出现故障的时候，能用备用的服务器顶替原服务器上线使用；在主服务器的存储出现故障或者数据库用 RMAN 恢复的宕机时间无法忍受的时候，Standby 数据库能够及时顶替上线，使业务可以持续进行。

这种利用 Oracle Rman 的备份和 Standby 容灾相结合，可在各种情况下灵活的选择恢复方式，使数据库下线时间缩短到最小，尽量减少业务损失。

#### 参考文献

1. 杨晓红. 信息系统容灾技术的分析与研究 计算机工程与设计 2005, 26 (10); 2727-2729.
2. 陈惠君. 远程灾备系统中光纤网络故障实时监控的实现 计算机工程 2005, 31 (增刊); 270-2