

刘晓东 (中国广播电影电视节目交易中心)

AIX 下的系统及数据备份

Backup of System and Data in AIX

摘要: 在较为大型的计算机应用系统中, 系统及数据的备份变得越来越重要, 它是整个系统安全的一个重要组成部分, 也是核心部分, 如何进行数据的日常备份, 是每个系统管理员必须要考虑的问题。本文结合实际工作经验, 以 IBM AIX 操作系统及 DB2 数据库为例, 介绍系统及数据备份的方法。

关键词: AIX 备份

1 数据备份

(1) 数据备份包括: 操作系统备份、文件系统备份和用户数据备份。

(2) 备份策略流程图。从策略上建议使用图1所示备份流程:

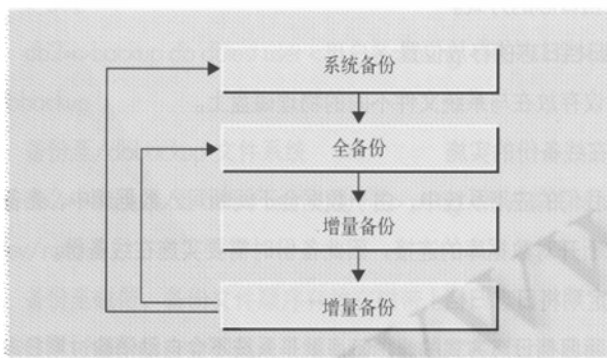


图1 备份流程

2 操作系统及文件系统的备份策略及方法

2.1 备份策略

(1) 对操作系统的备份, 不需太频繁, 建议几个月到半年做一次。

```
# smitty mkysyb
```

或命令

```
# /usr/bin/mkysyb -i /dev/rmt0
```

操作系统备份是在系统发生致命故障的情况下对其进行恢复的主要依据, 因此备份带应由系统管理员妥善保管。

(2) 对重要文件系统的备份。建议先做一个全备份 (0级备份), 再每天做增量备份; 可直接使用命令或让系统定时周期性的来提交。表1是一个增量备份的规划表实例:

表1 增量备份规划表

Sun	Mon	Tue	Wed	Thur	Fri	Sat
1 level0	2 level5	3 level5	4 level5	5 level5	6 level5	7 level5
8 level0	9 level5	10 level5	11 level5	12 level5	13 level5	14 level5
15 level0	16 level5	17 level5	18 level5	19 level5	20 level5	21 level5
22 level0	23 level5	24 level5	25 level5	26 level5	27 level5	28 level5
29 level0	30 level5	31 level5				

每个月:

每周日做一次0级备份。

其他日期 (周1-周6) 做5级备份。

2.2 备份方法

可采用由系统管理员定期执行备份命令或由系统自动定期执行备份操作的方式。如需系统定期执行, 可按照下面的步骤执行:

(1) 在crontab文件中加入

```
0 23 * * 0 backup-0 -u -f /dev/rmt0 /filesystem1
```

```
0 23 * * 1-6 backup-5 -u -f /dev/rmt0.1 /filesystem1
```

或

```
0 23 * * 0 /etc/backup file1
```

```
0 23 * * 1-6 /etc/backup file2
```

crontab的命令格式为:

```
mins hour day_month month day_of_week command
```

```
0-59 0-23 1-31 1-12 0-6
```

(2) 分别编辑以下文件:

```
* /etc/backup -file1
```

```
* /etc/backup -file2
```

```
backup -5 -u -f /dev/rmt0.1 /filesystem1
```

(3) 对于网上其他机器的文件系统, 使用NFS来实现远程mount, 以解决文件系统的备份。

方法:

手动使用exportfs和mount命令来远程mount文件系统。

2.3 文件系统的恢复

(1) 方法1: 恢复整个文件系统使用命令:

```
# restore -rvqf /dev/rmt0
```

(2) 方法2: 恢复备份带上的第三个备份数据和第五个备份数据:

```
# restore -xvqs 3 -f /dev/rmt0.1
```

```
# restore -xvqs 2 -f /dev/rmt0
```

(3) 方法3: 恢复某一单独的文件:

```
# restore -xvqf /dev/rmt0 file1
```

(4) 查看备份带上的内容:

```
# restore -Tvrf /dev/rmt0
```

2.4 重要文件的备份与恢复

对于重要文件可使用tar命令来对单个文件或目录进行备份和恢复。

3 用户数据的备份策略及方法

3.1 数据库的备份/恢复

对于数据库的备份不要使用操作系统命令做数据备份, 否则在恢复数据时可能会导致大范围的数据库损坏。

在用户制定备份/恢复策略中应考虑以下一些事项:

3.1.1 考虑要使用的日志类型

日志文件是由DB2用来确保数据库完整性, 为了确保用户数据的完整性, DB2实施提前写日志存档模式。DB2提供两种日志类型: 循环日志与归档日志。

与循环日志相比, 归档日志可以较完整地恢复用户已提交的事

务。但要求数据库管理员清楚地了解用户数据库的结构, 并熟悉所有备份/恢复命令。

用户可以根据数据库的物理设计和提供的服务类型, 选择使用哪种类型的日志。

3.1.2 决定哪一级别的数据备份/恢复

用户可以根据数据库的物理设计, 确定在执行备份/恢复操作时, 是在数据库级别还是表空间级别。

3.1.3 要意识到恢复动作可能包括前滚操作

此时要求数据库的日志类型必须是归档日志。

3.1.4 要清楚必须进行前滚操作的时间点

管理员根据数据库发生故障的时间点决定前滚至当前日志路径目录中所能找到的所有日志文件的末尾还是至用户最后一个提交的事务的时间点。

3.2 循环日志

循环日志是操作系统缺省的日志类型, 不支持在线备份, 备份时需断开数据库连接, 所以在很多情况下不太适用, 其优点是日志循环记录的, 日志文件不会无限制增大, 备份及恢复方法相对较简单。

3.3 归档日志

3.3.1 归档日志方式的特点

根据备份文件及归档日志, 可以使用户将数据恢复至灾难发生前最后提交的事务的时间点, 将用户的损失减少到最小。因此我们采用的是归档日志的方式。

3.3.2 归档日志的存放位置

建议存放在与系统文件不同的物理磁盘上。

3.3.3 在线备份的实施

在我们的应用系统中, 由于数据会不间断写入数据库中, 在备份时不能断开对数据库的连接, 因此备份时需要实施在线备份。

3.3.4 定期将日志归档

采用归档日志方式的一个特点就是系统不会自动消除过期日志, 需要管理员手动将这些日志清除或存放在其他介质上。

3.3.5 妥善保存备份介质

备份介质(磁带或光盘)应妥善保存。

3.3.6 备份时间安排

根据业务情况, 可以定期、不定期对数据进行备份。

3.4 归档日志类型的数据库备份与恢复

采用该方法的优缺点如下:

优点为:

支持数据库在线备份, 并可执行表空间级别的备份与恢复, 节省工作时间;

可将数据库前滚恢复至发生故障前用户做最后一次提交

(commit)操作的时间点的状态。

缺点为:

实施方法相对复杂。

3.5 归档日志备份方法的具体实施方法

3.5.1 更改日志类型——由循环日志更改为归档日志

如果决定采取归档日志方式,那么首先要将日志类型更改为归档日志,具体做法如下:

以DB2用户登录后,执行以下命令:

```
$ db2 update db cfg for was using LOGRETAIN ON
$ db2 update db cfg for <数据库名称> using LOGRETAIN ON
$ db2 stop
```

\$ db2start 重新启动实例

以后不用再执行此命令

3.5.2 备份数据库

我们以名为dbtest的数据库为例来进行介绍

创建存放备份介质的文件系统 如: /dbbackup

smitty crfs

* 更改/dbbackup文件系统的属性

chown-R <用户名.用户组名> /dbbackup

chown-R 755 /dbbackup

* 备份dbtest数据库

db2

```
db2=>backup db dbtest user <用户名> using <用户口令> online to
/dbbackup
```

备份至/dbbackup 文件系统

```
db2=>backup db dbtest user <用户名> using <用户口令> online to
/dev/rmt0.1
```

备份至磁带,备份文件顺序存放在磁带上

注意:备份后应立刻记录下所备份的数据库名、设备名和备份时间戳,留待恢复数据库时使用。

3.5.3 恢复数据库

* 由存放在/dbbackup 文件系统工程备份文件恢复dbtest数据库

db2

```
# db2=> restore db dbtest user <用户名> using <用户口令> from /
dbbackup taken at <备份时间戳> to /newfs [into <new-db-name>]
```

/newfs:恢复后的dbtest数据库的存放位置,[]中信息为将dbtest数据库恢复至一个新数据库,可根据实际情况选择使用。

* 由存放在磁带上的备份文件恢复chnpec数据库

db2

```
db2=>restore db dbtest user <用户名> using<用户口令> from /
dev/rmt0.1 taken at <备份时间戳> to /newfs [info <new-db-name>]
```

3.5.4 前滚日志文件

* 查看有关前滚日志文件的信息

```
db2=>rollforward db dbtest QUERY STATUS
```

* 前滚dbtest数据库

```
db2=>rollforward db dbtest to end of logs and complete overflow
log path <存放dbtest 数据库日志文件的目录>
```

overflow参数指定存放dbtest数据库日志文件的目录

注意:若更改了系统时间,如将系统时间调快5小时,可能会出现备份文件的时间戳比日志所记录的最后提交事物时间晚,从而导致前滚操作失败。此时需要手工归档日志,然后再做备份,步骤如下:

* 更改系统时间为正确时间

* 执行手工归档日志操作

```
db2=>archive log for db dbtest user <用户名>using<用户口令>
```

* 执行在线备份操作

* 之后对数据库执行的操作均记入新的日志中

* 恢复数据库

* 前滚数据库到日志末尾并终止前滚挂起

* 数据库处于正常状态

其他说明:对于脱机备份,可以在恢复数据库时使用without rolling forward参数,使恢复后的数据库直接可用。

```
db2 =>restore db dbtest user<用户名> using<用户口令>from /
dbbackup taken at<备份时间戳>to /newfs [into <new-db-name>] with
out rolling forward
```

