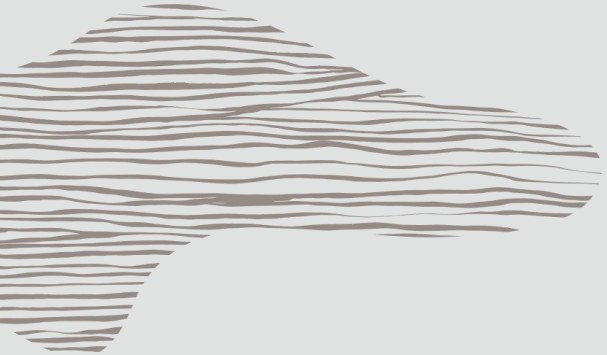
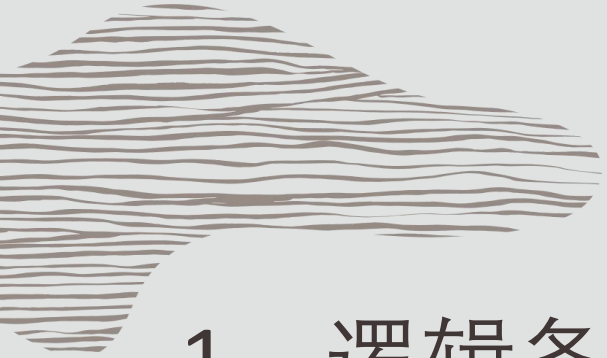


Oracle 备份恢复

2022年3月

官云龙

数据库备份



1、逻辑备份

2、物理备份

3、数据泵备份

备份就是数据库信息的一个拷贝

对于oracle而言，这些信息包括控制文件、数据文件以及重做日志文件等。

数据库备份的目的是为了防止意外事件发生而造成数据库的破坏后恢复数据库中的数据信息。

备份和恢复是两个互相联系的概念，备份就是将数据信息保存起来；而恢复则是当意外事件发生或者某种需要时，将已备份的数据信息还原到数据库系统中去。

备份和恢复计划

1）根据生产环境的恢复周期，制定详细的备份计划，然后严格执行

2）对备份，要在一定的时间内利用测试环境，进行故障恢复的练习

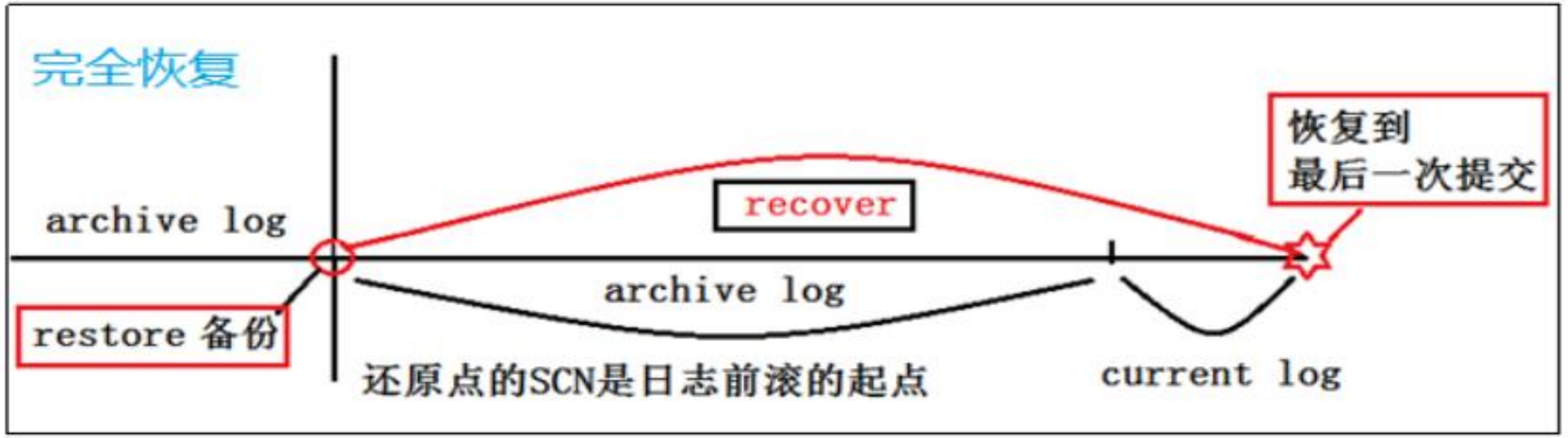
备份恢复分类
1）逻辑备份与恢复-- 面向object
①传统的导入导出：exp/imp：
②数据泵导入导出：expdp/impdp
逻辑备份就是热备数据库对象某一时刻状态，逻辑备份的恢复就是还原备份，没有recover的概念。
2）物理备份与恢复-- 面向datafile
①手工备份与恢复，也叫用户管理的备份与恢复，通过OS 的命令，完成备份与还原，然后再运用日志进行恢复。
②自动备份与恢复，利用oracle 的备份恢复工具RMAN，使还原与恢复过程自动化程度较高，可以备份恢复ASM FILE。

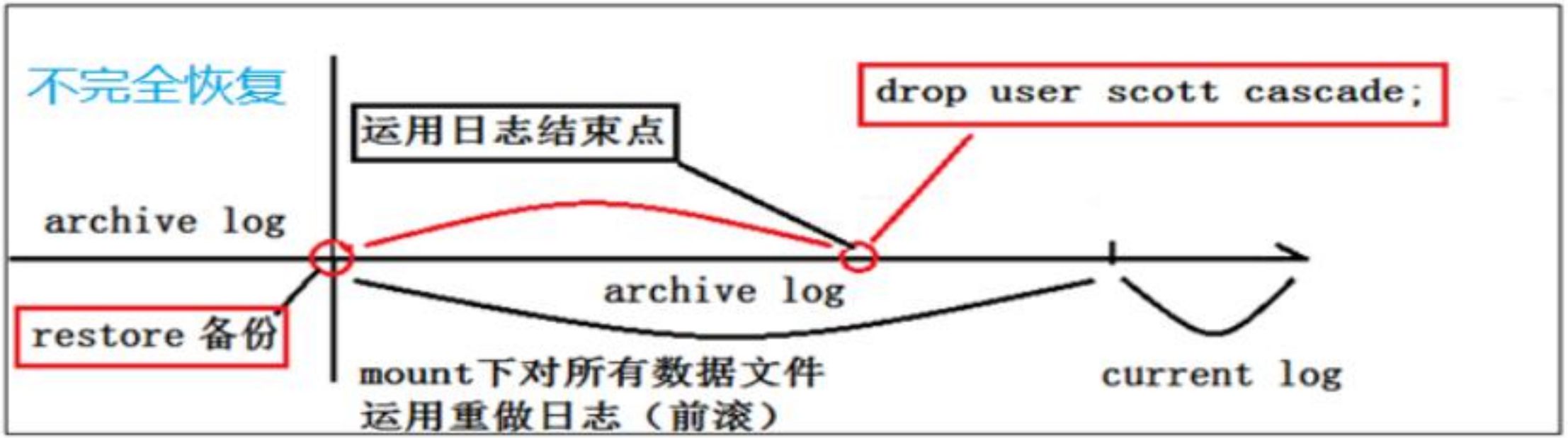
完全恢复与不完全恢复
media failure后，需要运用日志进行recover。

1）完全恢复：

利用完整备份或部分备份，可以将datafile恢复到failure前得最后一次commit，不会出现数据丢失。

2）不完全恢复
需要运用完整备份和日志将database恢复到过去的某个时间点（或SCN），有数据丢失。


完全恢复
archive log
recover
恢复到
最后一次提交
archive log
restore 备份
还原点的SCN是日志前滚的起点
current log


不完全恢复
运用日志结束点
drop user scott cascade;
archive log
archive log
restore 备份
mount下对所有数据文件
运用重做日志（前滚）
current log

归档与非归档

1）归档模式：redo log 写入 archive log

2）非归档模式：没有archive log, redo log file循环覆盖

	一致性备份（冷备）	不一致备份（热备）	完整备份	不完整备份
归档模式	✓	✓	✓	✓
非归档模式	✓	✗	✓	✗

手工备份与恢复

一）相关命令
备份和还原都使用OS命令(数据库是文件系统)，如linux中的cp

二）备份前进行检查:
1）检查需要备份的数据文件
SQL> select name from v$datafile;
SQL> select file_id,file_name,tablespace_name from dba_data_files;

2）检查要备份的控制文件
SQL> select name from v$controlfile;

3）在线redo日志可以不做备份

三）dbv检查坏块
在手工备份前，应该检查datafile 是否有坏块，备份完后对备份也要做检查。
对某个datafile做坏块检查
$ dbv file=/u01/oradata/prod/users01.dbf feedback=50
DBVERIFY - 开始验证: FILE = /u01/oradata/prod/users01.dbf

四）冷备的注意事项：

1）必须干净的关闭数据库，以保证数据一致性。

SQL>shutdwon immediate;

2）在OS下必须备份所有数据文件(完整备份)

3）在OS下必须备份控制文件(至少备份一个）

4）非归档备份还原策略

恢复时还原所有备份。

SQL>startup mount

SQL>alter database clear logfile group n;（n为所有在线日志组）

SQL>alter database open;

五）手工非一致性备份（热备份）

1）在备份前要进入热备模式，备份后要结束热备模式

执行begin backup 设置备份模式（在数据文件上生成检查点，写入scn ， 将来恢复的时候以此scn为起点）

对只读的表空间不能做热备份，临时表空间不需要备份，特别强调：NOARCHIVE模式下不支持手工热备。

对整个数据库设置热备模式：SQL> alter database begin backup;

对整个数据库结束热备模式：SQL> alter database end backup;

对单个表空间设置热备模式：SQL> alter tablespace users begin backup;

对单个表空间结束热备模式：SQL> alter tablespace users end backup;

手工恢复

一）基本概念

1）完全恢复的步骤

1）restore: OS拷贝命令还原所有或部分datafile

2）recover：SQL*PLUS利用归档日志和当前的redo日志做恢复

2）完全恢复可以基于三个级别

recover database：　所有或大部分datafile损坏，一般是在mount状态完成

recover tablespace：　非关键表空间损坏，表空间下某些数据文件不能访问，一般是在open下完成

recover datafile：　单一或少量数据文件损坏，可以在mount或open 状态完成

3）什么是关键文件

如果关键文件损坏，数据库将不能维持在open状态，或崩溃或死机!

哪些文件是关键文件：①system01 file，②undotbs file，③control file，④current log file

4）恢复过程可以查看的视图：

1）v$recover_file　查看需要恢复的datafile

2）v$recovery_log　查看recover 需要的redo 日志

3）v$archvied_log　查看已经归档的日志

适用场景

1）recover database (所有或大部分数据文件损坏，mount或open下进行)

OS：使用cp 还原受损的dbf（不一定是全部，v$recover_file记录的都需要还原）

SQLPLUS：

①recover database;

②alter database open;

2）recover tablespace (针对表空间的非关键数据文件损坏，一般是open下进行

OS:使用cp 还原该表空间XXX下的所有数据文件

SQLPLUS：

①alter tablespace XXX offline immediate;

②recover tablespace XXX;

③alter tablespace XXX online;

3）recover datafile （单个或几个数据文件损坏，关键文件在mount下进行，非关键文件在open下进行)

第一种情形

OS:使用cp 还原相关的关键数据文件(mount)

SQLPLUS：

①recover datafile 6,8;

②alter database open;

第二种情形

OS:使用cp 还原相关的非关键数据文件(open)

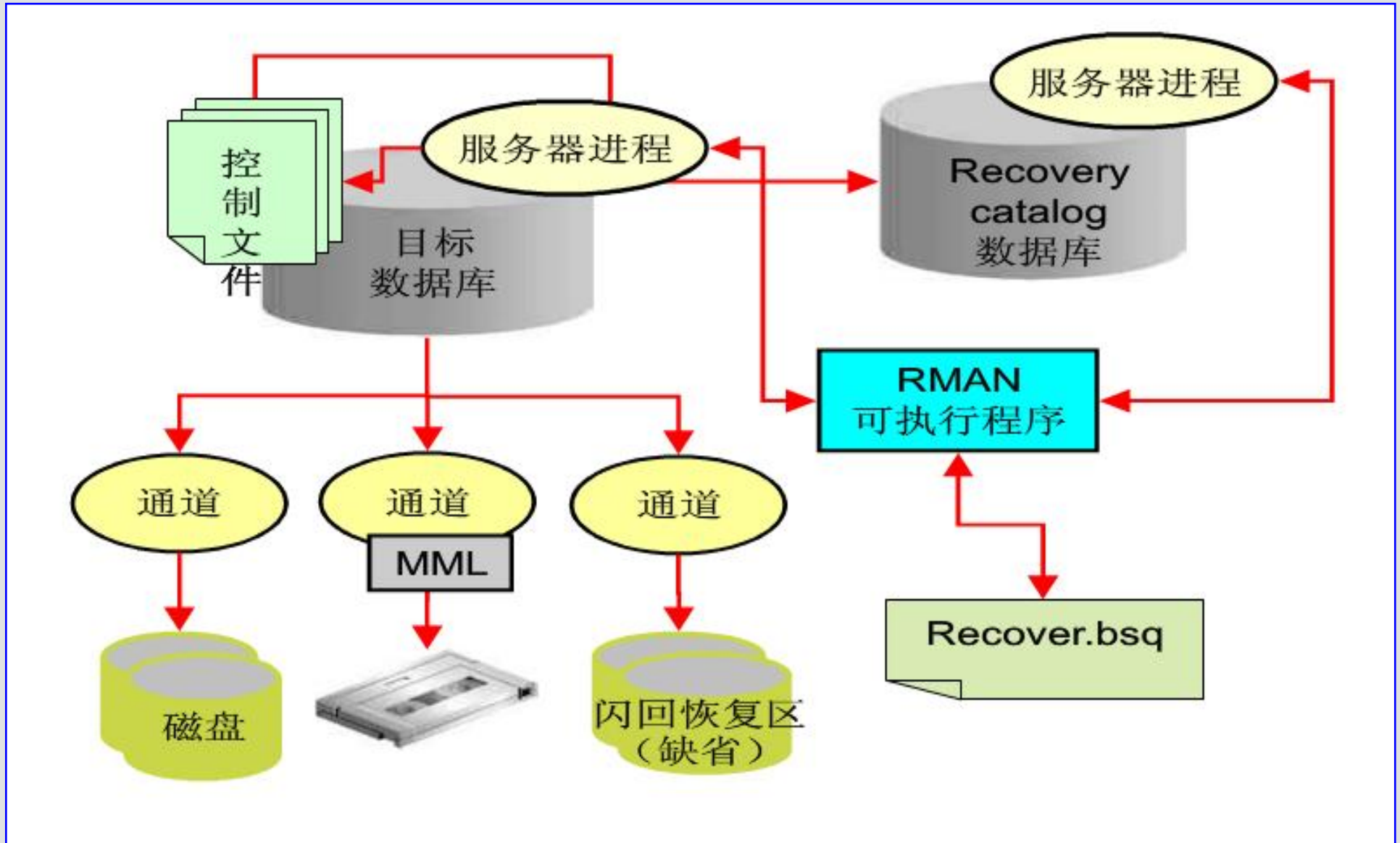
SQLPLUS：

①alter database datafile 6,8 offline;

②recover datafile 6,8;

③alter database datafile 6,8 online;

RMAN备份与恢复


服务器进程
控制文件
服务器进程
目标数据库
Recovery catalog 数据库
RMAN 可执行程序
通道
通道
MML
通道
Recover.bsq
磁盘
闪回恢复区（缺省）

1、功能：
1）Recovery MANager，是备份和恢复数据库的管理工具

2）由server process进行备份和恢复

3）rman 备份的文件种类
①datafile（database、tablespace、datafile）

②controlfile、spfile

③archivelog

4）在归档模式下支持非一致性备份（热备）
非归档方式的RMAN只能冷备，并在mount下做，但手工备份在mount下cp出来的备份对于RMAN是不可用的。
非归档方式的RMAN恢复只能还原最后一次备份。

2、特点:

1）不备份数据文件中未分配的块，以及未使用的块。节省时间和空间。

2）备份时自动检查数据文件是否有坏块，因为RMAN是ORACLE BLOCK级备份技术

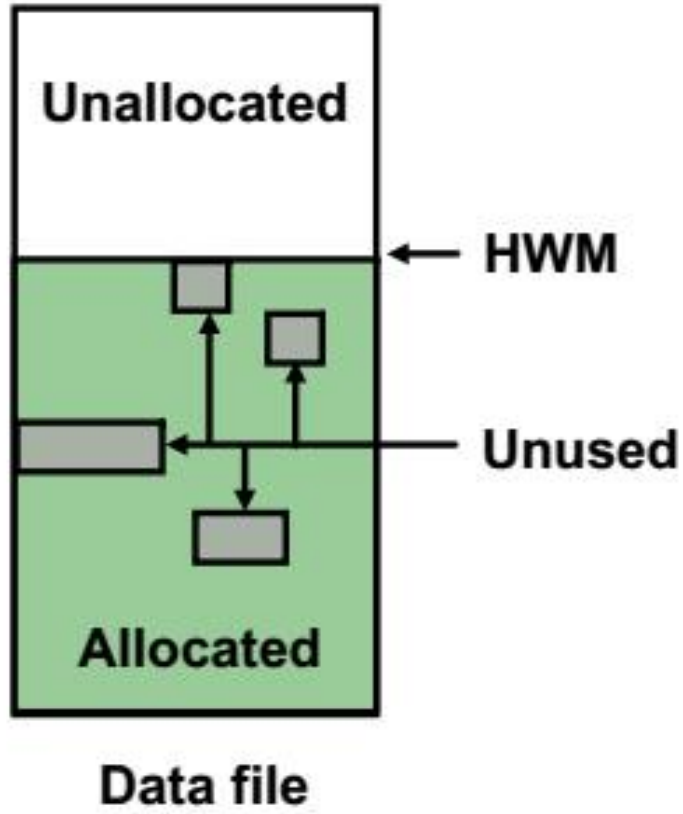
3）不备份临时表空间

4）可以实现增量备份

5）支持多通道并行工作

RMAN 备份可以忽略两种block

- Unallocated（未分配的）
- Unused（未使用的）



RMAN的命令格式：

1）交互式（也叫stand alone方式）

```
RMAN> shutdown immediate;
RMAN> startup force mount;
RMAN> alter database open;
RMAN> alter system switch logfile;
RMAN> select * from scott.emp;
```

2）批处理方式（也叫job方式）

```
RMAN>run {
shutdown immediate;
startup mount;
allocate channel c1 type disk;
allocate channel c2 type disk;
backup database format '/u01/myrman/%d_%s.bak';
alter database open;
release channel c1;
release channel c2;
}
```

RMAN备份功能

一）backupset 备份集

1）完整备份：

①RMAN> backup database format='/u01/myrman/prod_%s.bak' ;

②RMAN> backup database plus archivelog delete all input;

说明：备份所有数据文件及控制文件、spfile文件与所有归档日志，并删除旧的归档日志，当delete all input后，控制文件相关信息（v$archived_log）也会被更新。archivelog物理日志也被删除。

③RMAN> backup database format '/u01/myrman/%s_bak' plus archivelog delete input skip inaccessible; (不推荐使用)

说明：backup database 紧接format可以使datafile 的备份片指定到format的目的地，archivelog的备份是根据控制文件中（v$archived_log）中的内容导航的，如果控制文件中记录了而实际归档中又不存在，则会报错，skip inaccessible的含义是跳过物理上缺失的日志文件。

④RMAN> backup as compressed backupset incremental level 0 database；0级增量备份（压缩），省略differential描述, 默认的就是差异增量备份

⑤RMAN> backup cumulative incremental level 1 database；1级累计增量备份

因为有了全备，随时可以还原备份，还原点之前的归档日志一般没有就什么用处了，如果想单独还原归档日志备份可以使用：

restore archivelog all;

2）备份表空间：

RMAN> backup tablespace users format '/u01/myrman/users_%s.bak' tag=userbak;

RMAN> backup tablespace system plus archivelog delete all input；备份指定表空间及归档日志，并删除旧的归档日志

3）备份数据文件

RMAN> backup datafile 3,5 format '/u01/myrman/%d_%s.bak'；备份数据文件，可以多个，以“，”分开。

4）备份归档日志

RMAN> backup archivelog all delete input;

备份数据文件，控制文件，spfile及归档日志，然后删除所有归档原始文件

delete input和all delete input的区别：如果归档原始文件有多个路径的副本，前者仅删除备份了的归档原始文件。

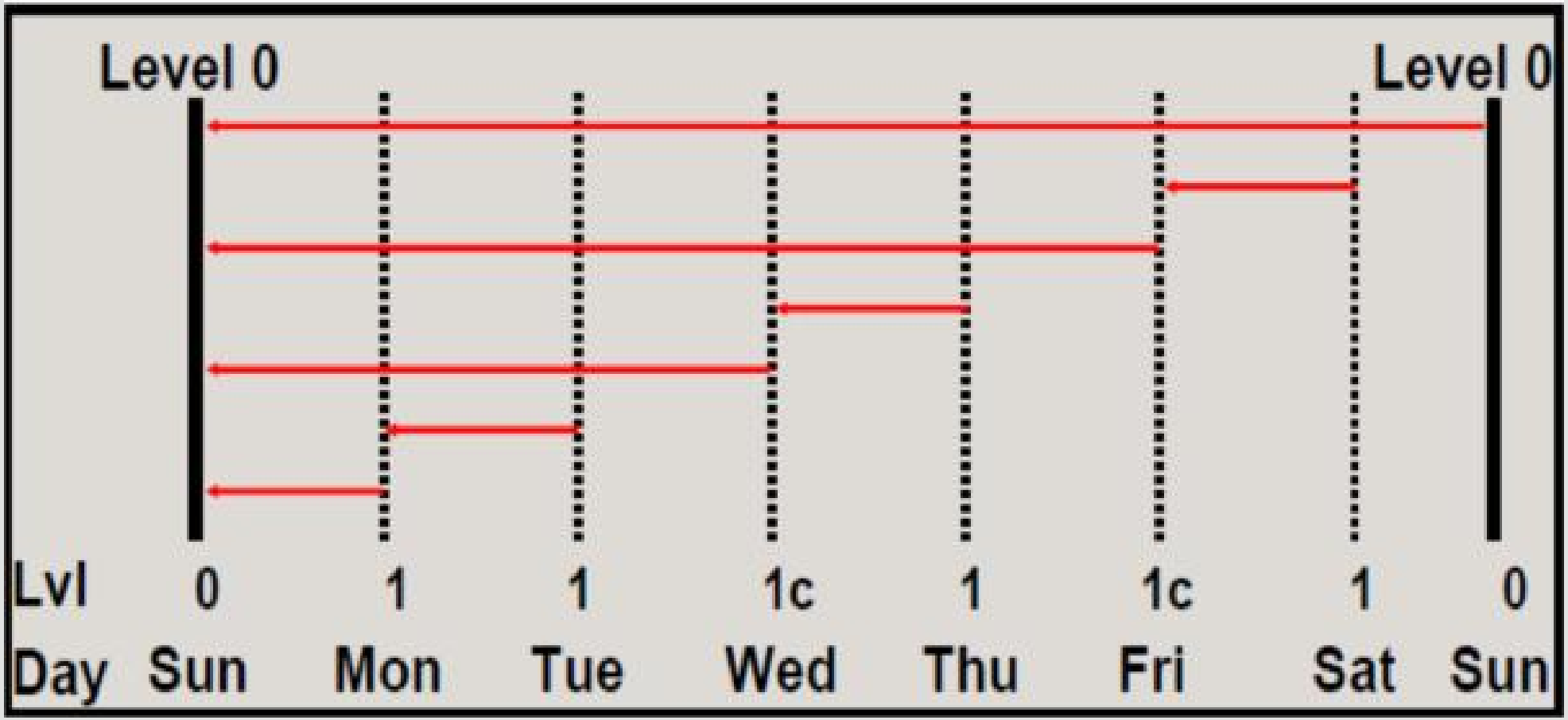
5）备份控制文件

RMAN> backup current controlfile;

两种增量策略:

①差异增量备份（Differential incremental backup）：以某次以来同级别或低级别的备份作为基础备份

②累积增量备份（Cumulative incremental backup)：以某次以来比自己级别低的备份作为基础备份



RMAN完全恢复

一）概念

1）recover 恢复：

①归档：完全恢复和不完全恢复

②非归档：只能恢复到最后一次备份状态（还原）

2）完全恢复：

先对数据库做一个备份（如果是archived模式就做热备）

换一种形式，我们将下面的run{}写到linux的脚本中，叫做/u01/oradata/prod/myrman.sh

```
run {
allocate channel c1 type disk;
allocate channel c2 type disk;
backup database format '/u01/myrman/%d_%s.bak';
release channel c1;
release channel c2;
}
```

完全恢复范例

范例1：system、users、undo表空间损坏

1）在线删除

2）模拟关键表空间损坏，然后启动数据库，报system01.dbf读写错误

```
[oracle@prod ~]$ rm /u01/oradata/prod/system01.dbf
[oracle@prod ~]$ rm /u01/oradata/prod/undotbs01.dbf
[oracle@prod ~]$ rm /u01/oradata/prod/users01.dbf
```

使用run{}恢复

```
run{
startup force mount;
restore datafile 1,3,4;
recover datafile 1,3,4;
alter database open;
}
```

范例2：恢复表空间（数据库open状态）。

1）环境
因非关键数据文件介质损坏，需要将其表空间恢复到一个新的物理位置。
create table scott.t1(id int);
insert into scott.t1 values(1);
commit;
2）模拟users表空间损坏，删除数据文件。
[oracle@prod ~]$ rm /u01/oradata/prod/users01.dbf
3）清除db buffer，查证访问数据文件失败
SQL> alter system flush buffer_cache；清除后session要重新登录
4）建个新目录
假设介质损坏了，更换需要时间，先把数据文件恢复到一个新的目录下（不同的物理位置）
[oracle@prod ~]$ mkdir /u01/oradata/prod/dir1
5）使用RMAN恢复表空间
RMAN>run{
alter database datafile 4 offline;
set newname for datafile 4 to '/u01/oradata/prod/dir1/users01.dbf';
restore tablespace users;
switch datafile 4;
recover tablespace users;
alter database datafile 4 online;
}

说明:
①set newname for 告诉RMAN还原数据文件的新位置在哪里。这个命令在restore前出现。
②switch datafile 更新controlfile，让控制文件使用这个新位置恢复数据。这个命令要在recover前出现。
5）验证

```
SQL> select * from scott.t1;
        ID
----------
         1
```

6）待介质更换完成后，可以将表空间迁移回原来位置。

```
SQL> alter tablespace users offline;
[oracle@prod~]mv /u01/oradata/prod/dir1/users01.dbf /u01/oradata/prod
SQL>alter tablespace users rename datafile '/u01/oradata/prod/dir1/users01.dbf' to '/u01/oradata/prod/users01.dbf';
```

更新控制文件和数据字典

```
SQL> alter tablespace users online;  有时候需要recover datafile 4一下
```

7）再验证

```
SQL> select * from scott.t1;
```

RMAN不完全恢复

一）概念：

RMAN不完全恢复的三个标准模式：基于time、基于scn和基于sequence：

与手工不完全恢复相比原理类似，语法稍有不同：

	手工不完全恢复	RMAN不完全恢复
基于time	until time XXX	until time XXX
基于scn	until change XXX	until scn XXX
基于日志	until cancel	until sequence XXX

范例：假定参数文件，控制文件，数据文件以及在线(当前)日志全部损毁了，有备份（热备）和归档，如何恢复数据库。

1）环境

```
SQL> select * from v$tablespace;
      TS# NAME                           INC BIG FLA ENC
--------- ------------------------------ --- --- --- ---
        0 SYSTEM                         YES NO  YES
        1 SYSAUX                         YES NO  YES
        4 USERS                          YES NO  YES
        6 EXAMPLE                        YES NO  YES
        8 TEST                           YES NO  YES
        3 TEMP                           NO  NO  YES
        2 UNDOTBS1                       YES NO  YES
SQL> select owner,table_name,tablespace_name from dba_tables where table_name='T1';
OWNER                          TABLE_NAME                     TABLESPACE_NAME
------------------------------ ------------------------------ ------------------------------
SCOTT                          T1                             USERS
SQL> select * from scott.t1;
        ID
----------
         1
```

2）关闭数据库，然后删除参数文件、数据文件、控制文件、在线日志
SQL> shutdown abort
[oracle@cuug ~]$] cd /u01/oracle/dbs
[oracle@prod dbs]$ rm spfileprod.ora
[oracle@prod dbs]$ rm initprod.ora
[oracle@prod dbs]$ cd /u01/oradata/prod
[oracle@prod]$ rm * 删除数据库（包括datafile,ctl,log）

3）RMAN恢复参数文件
[oracle@prod ~]$ rman target /
connected to target database (not started)
RMAN> startup nomount; 没有了参数文件，SQL*PLUS是无法启动实例的，但RMAN可以，所以startup nomount一定要在RMAN下做！！！
RMAN> restore spfile from '/u01/flash_recovery_area/prod/autobackup/2013_01_16/o1_mf_s_932826360_d76c7r4y_.bkp';
查看在dbs/目录下已经产生spfileprod.ora文件。证明spfile 恢复好了。
RMAN> startup force nomount 使用恢复的spfile启动
4)RMAN 恢复控制文件
RMAN> restore controlfile from
'/u01/flash_recovery_area/prod/autobackup/2013_01_16/o1_mf_s_932826360_d76c7r4y_.bkp';
RMAN> alter database mount; 加载恢复的控制文件
RMAN> list backup; 可以看到RMAN的元数据了
如果元数据不全，可以使用catalog注册缺失的元数据

5)还原所有的数据文件
RMAN> restore database;
6)运用日志做恢复
RMAN> recover database; 尝试完全恢复，使用尽量多的归档恢复。
RMAN> recover database until sequence n; “根据提示unable to find archived log archived log thread=1 sequence=n”
7)打开数据库
RMAN> alter database open resetlogs; 当前日志缺失，又使用备份的控制文件，所以是不完全恢复的形式打开数据库。
SYS@ prod>select * from scott.t1;
select * from scott.t1
*
ERROR at line 1:
ORA-00942: 表或视图不存在 当前日志缺失，丢失数据。