



华北ORACLE用户组

CNNOUG



ORACLE 快速备份恢复

杜诚文



CONTENTS

01

备份的重要性

02

传统备份恢复

03

备份优化

04

LVM快照技术

05

快速恢复

01

备份的重要性



备份的重要性

数据备份是容灾的基础，是指为防止系统出现操作失误或系统故障导致数据丢失，而将全部或部分数据集合从应用主机的硬盘或阵列复制到其它的存储介质的过程。传统的数据备份主要是采用内置或外置的磁带机进行冷备份。但是这种方式只能防止操作失误等人为故障，而且其恢复时间也很长。随着技术的不断发展，数据的海量增加，不少的企业开始采用网络备份。网络备份一般通过专业的数据存储管理软件结合相应的硬件和存储设备来实现。

备份的重要性

对数据的威胁通常比较难于防范，这些威胁一旦变为现实，不仅会毁坏数据，也会毁坏访问数据的系统。造成数据丢失和毁坏的原因主要如下几个方面。

1. 数据处理和访问软件平台故障。
2. 操作系统的设计漏洞或设计者出于不可告人的目的而人为预置的“黑洞”。
3. 系统的硬件故障。
4. 人为的操作失误。
5. 非法访问者的恶意破坏。
6. 供电系统故障等。

02

传统备份恢复



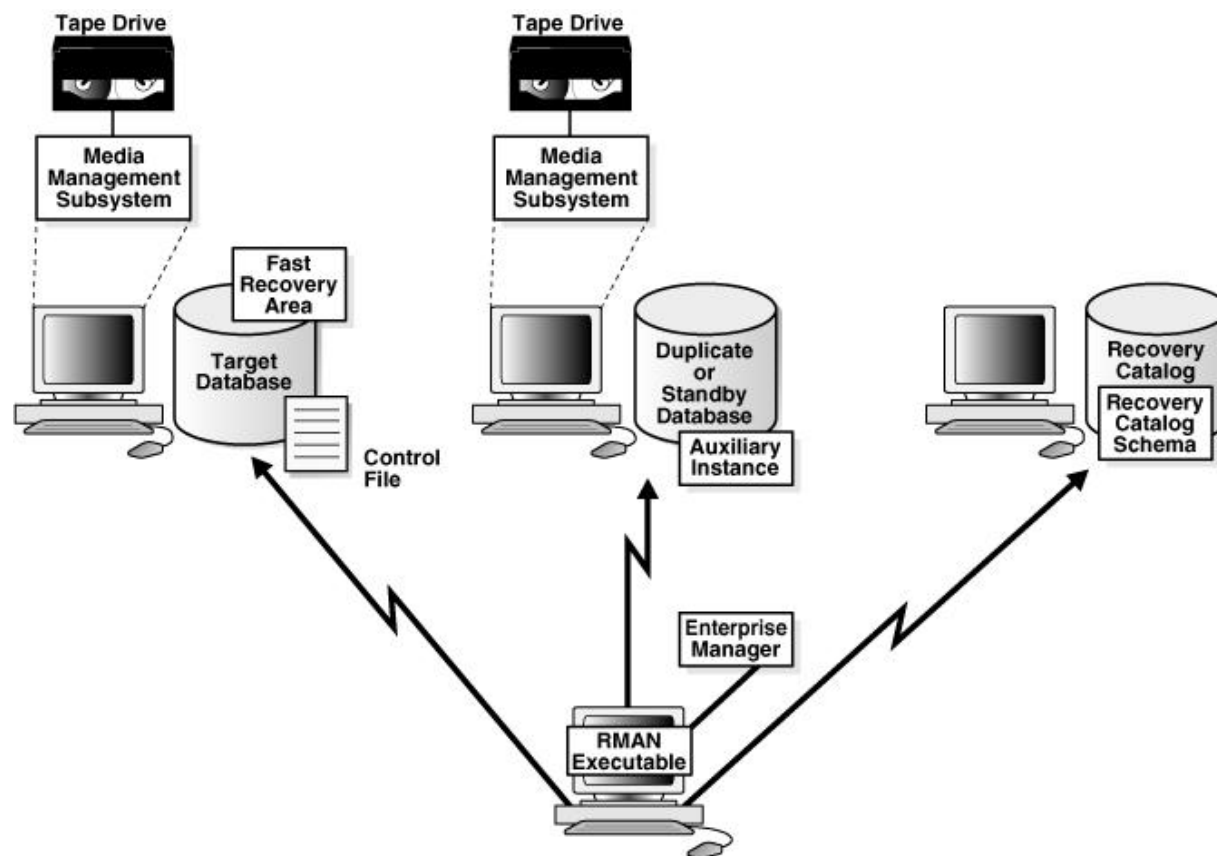
传统备份恢复

Feature	Recovery Manager	User-Managed	Data Pump Export
Closed database backups	Supported. Requires instance to be mounted.	Supported.	Not supported.
Open database backups	Supported. No need to use <code>BEGIN/END BACKUP</code> statements.	Supported. Must use <code>BEGIN/END BACKUP</code> statements.	Requires rollback or undo segments to generate consistent backups.
Incremental backups	Supported.	Not supported.	Not supported.
Corrupt block detection	Supported. Identifies corrupt blocks and logs in <code>V\$DATABASE_BLOCK_CORRUPTION</code> .	Not supported.	Supported. Identifies corrupt blocks in the export log.
Backup repository	Supported. Backups are recorded in the control file, which is the main repository of RMAN metadata. Additionally, you can store this metadata in a recovery catalog, which is a schema in a different database.	Not supported. DBA must keep own records of backups.	Not supported.
Backups to a media manager	Supported. Interfaces with a media management software. RMAN also supports proxy copy, a feature that enables a media manager to manage completely the transfer of data between disk and backup media.	Supported. Backup to tape is manual or controlled by a media manager.	Not supported.

RMAN备份

RMAN 备份支持的功能:

- Incremental backups
- Block media recovery
- Binary compression
- Encrypted backups
- Automated database duplication
- Cross-platform data conversion



RMAN恢复

需要进行数据恢复的场景：

Media Failures

- A media failure is a physical problem with a disk that causes a failure of a read from or write to a disk file that is required to run the database.

User Errors

- User errors occur when, either due to an error in application logic or a manual mistake, data in a database is changed or deleted incorrectly.

Application Errors

- Sometimes a software malfunction can corrupt data blocks.

传统数据库备份方案缺点

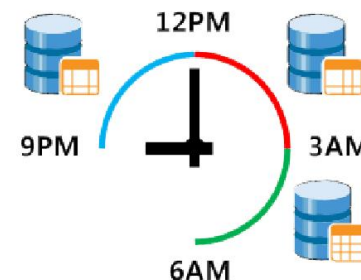
①存在数据丢失风险

丢失最近一次备份之后的数据



②备份窗口紧张

避开业务高峰
错开使用设备



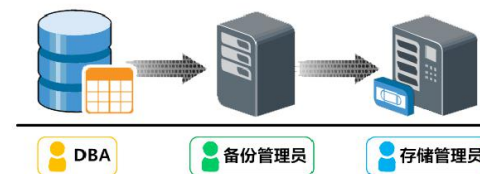
③恢复有效性无法验证

介质可靠性/恢复操作过程/恢复的时间



④备份系统复杂及碎片化管理

DBA不能确认备份体系可用



缺点

03

备份优化



快速备份

多级增量备份：

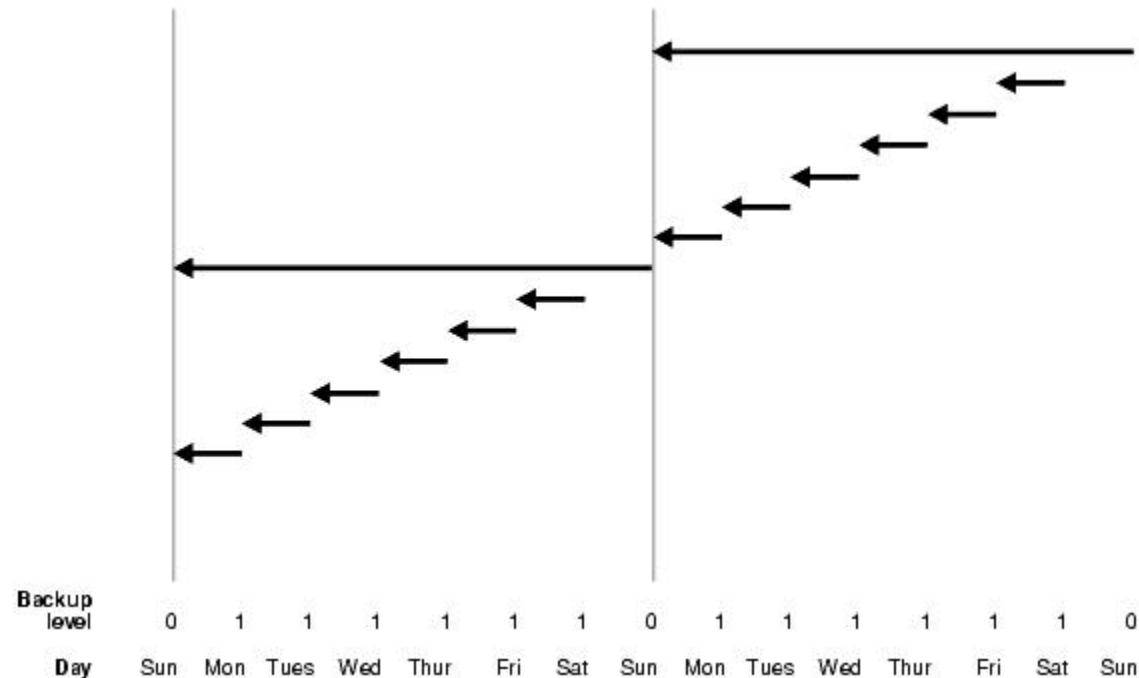
RMAN can create multilevel incremental backups. Each incremental level is denoted by a value of 0 or 1.

A **level 0 incremental backup**, which is the **base for subsequent incremental backups**, copies all blocks containing data. The only difference between a level 0 incremental backup and a full backup is that a **full backup is never included in an incremental strategy**. Thus, an incremental level 0 backup is a full backup that happens to be the parent of incremental backups whose level is greater than 0.

差异增量备份

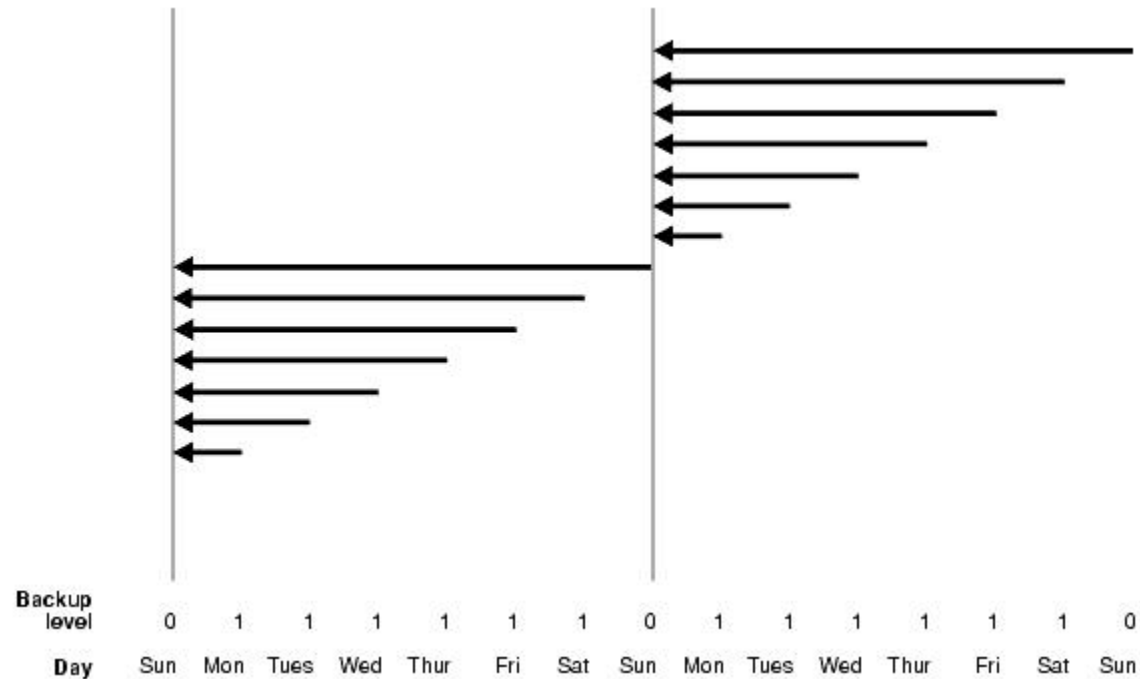
A level 1 incremental backup can be either of the following types:

A **differential incremental backup**, which backs up all blocks changed after the most recent incremental backup **at level 1 or 0**



累积增量备份

A **cumulative incremental backup**, which backs up all blocks changed after the most recent incremental backup **at level 0**



Block Change Tracking

The block change tracking feature for incremental backups improves incremental backup performance by recording changed blocks in each data file in a block change tracking file.

RMAN tracks changed blocks as redo is generated.

If block change tracking is enabled, then RMAN uses the change tracking file to identify changed blocks for incremental backups, thus **avoiding the need to scan every block in the data file.**

```
SQL> ALTER DATABASE ENABLE BLOCK CHANGE TRACKING USING FILE '/DB1/bct.ora';
```

```
SQL> ALTER DATABASE DISABLE BLOCK CHANGE TRACKING;
```

ORACLE 10G BLOCK CHANGE TRACKING INSIDE OUT (文档 ID 1528510.1)

Incrementally Updated Backups

RMAN enables you to **apply level 1 incremental backups to an older image copy of your data files**. You can roll forward the copy to the point in time of the most recent level 1 incremental backup. All **blocks changed since the image copy was created are overwritten with their new contents as of the time of the last level 1 incremental backup**.



Image Copy

An **image copy** is an exact copy of a single data file, archived redo log file, or control file. Image copies are **not stored in an RMAN-specific format**. They are identical to the results of copying a file with operating system commands. RMAN can use image copies during RMAN restore and recover operations, and you can **also use image copies with non-RMAN restore and recovery techniques**.

ZDLRA利用了RMAN的“incremental-forever”的策略可以在统一的存储设备上保存和管理多个Oracle数据库备份。利用持续的压缩、去重和数据块级别校验等技术来及时构建虚拟的数据库全备份。这个虚拟的数据库全备份其实就是一个数据库在一个时间点上的唯一镜像。

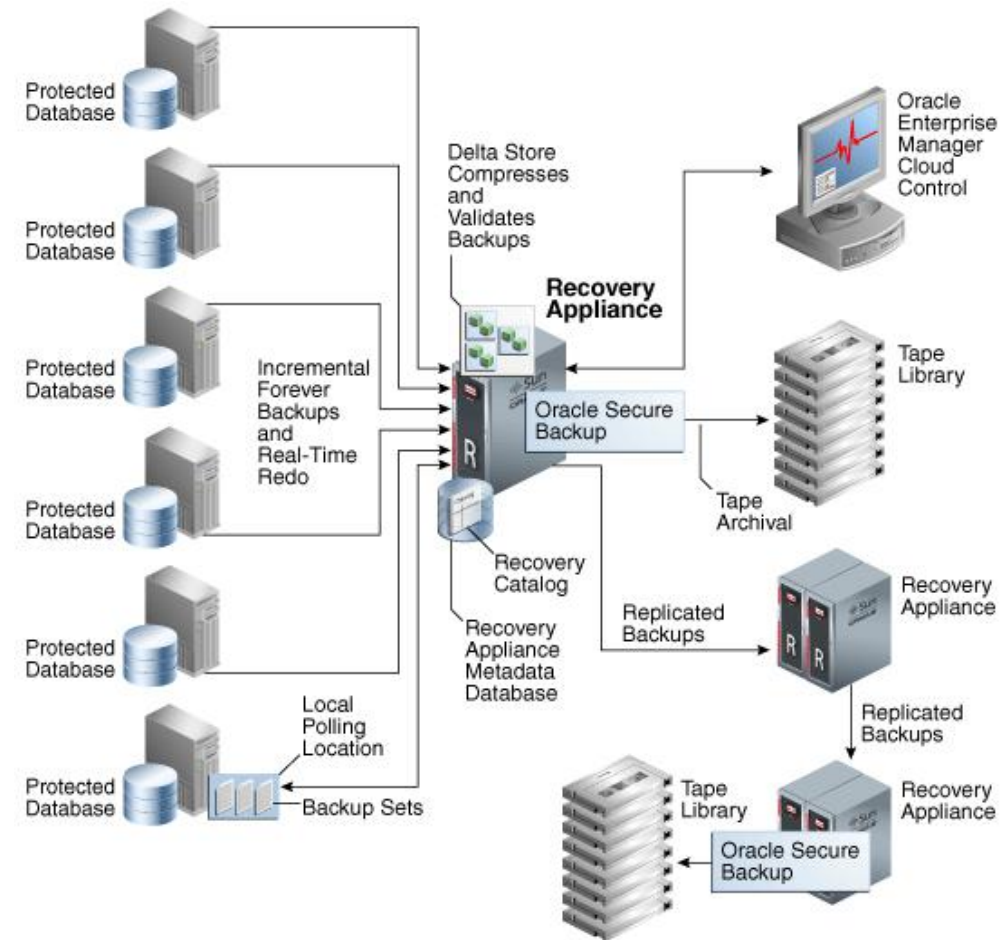


Image Copie

```
RMAN> BACKUP AS COPY INCREMENTAL LEVEL 0 TAG 'SRCDB-  
IMAGE' DATABASE FORMAT '/oradata/srcdb_img/SRCDB-LVL0-%U';
```

```
RMAN> BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF TAG  
'SRCDB-IMAGE' DATABASE FORMAT '/oradata/srcdb_img/SRCDB-  
LVL1-%U' PLUS ARCHIVELOG;
```

```
RMAN> RECOVER COPY OF DATABASE WITH TAG 'SRCDB-IMAGE';  
RMAN> VALIDATE COPY OF DATABASE;
```

04

LVM快照技术



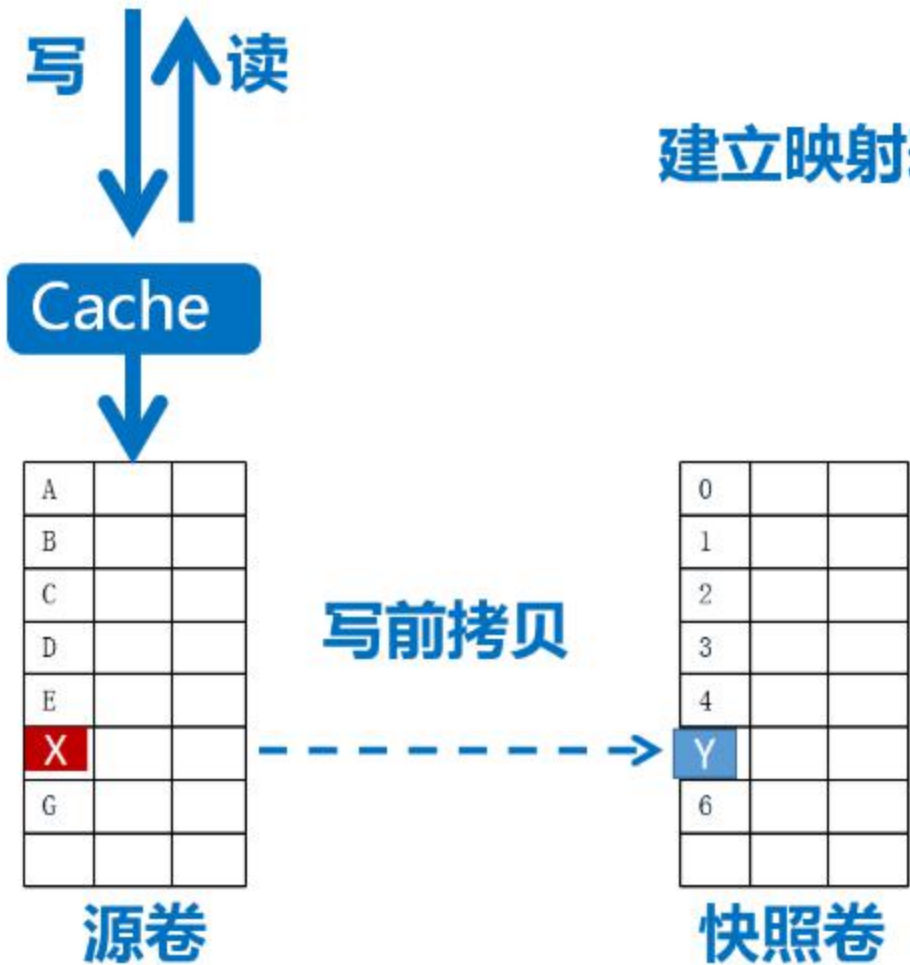
快照(Snapshot)技术

SINA对于快照的定义是：关于指定数据集合的一个完全可用拷贝，该拷贝包括相应数据在某个时间点(拷贝开始的时间点)的映像。快照可以是其所表示的数据的一个副本，也可以是数据的一个复制品。

快照的实现方式均由各个厂商自行决定，但主要技术分为2类，一种是写时拷贝COW(Copy On Write)，另一种，是写重定向ROW(Redirect On Write)。

写时拷贝COW

创建快照以后，如果源卷的数据发生了变化，那么快照系统会首先将原始数据拷贝到快照卷上对应的数据块中，然后再对源卷进行改写。



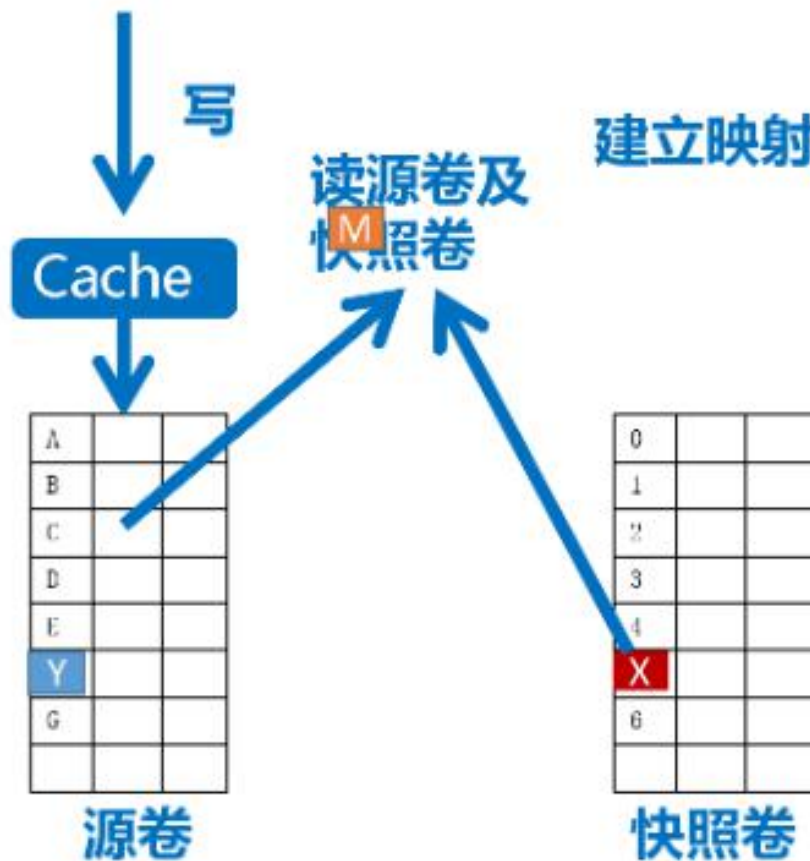
建立映射表：源地址<=>快照地址

source_add	snap_add
F	5

写数据：一共写2次：
源 ==> 快照
数据 ==> 源

写时重定向ROW

ROW(Redirect-on-write)，也称为写时重定向。把对数据卷的写请求重定向给了快照预留的存储空间，直接将数据写入快照卷。读时，创建快照前的数据从源卷读，创建快照后的，从快照卷读



建立映射表：源地址 \leftrightarrow 快照地址

source_add	snap_add
F	5

写数据：一共写1次：
数据 \Rightarrow 快照卷
读数据：查映射表，
读取重定向

两种技术对比

快照原理---两种技术对比



	COW	ROW
创建	耗时很短，创建时，占用少量存储空间	
写源卷	写性能受影响	写性能不受影响
读源卷	读性能不受影响	读性能受影响
读快照	读重定向，有轻微时延	
删除快照	瞬间完成	需要数据回拷到源卷，耗时可能很久
应用场景	读密集型系统	写密集型系统
ROW 与 COW 最大的不同就是： COW 的快照卷存放的是原始数据，而 ROW 的快照卷存放的是新数据		

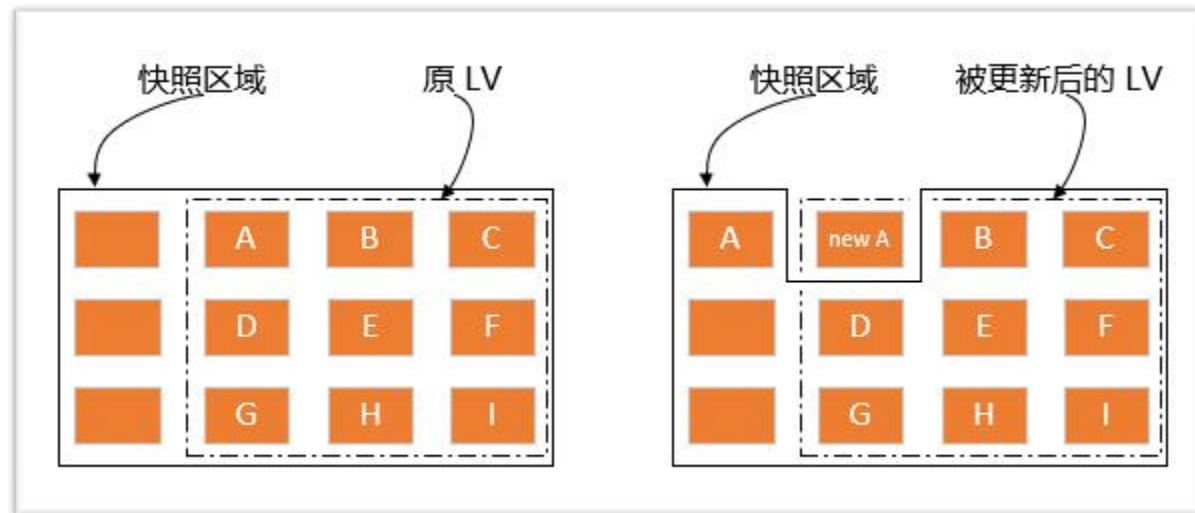
LVM快照技术

LVM 机制还提供了对 LV 做快照的功能，也就是说可以给文件系统做一个备份，这也是设计 LVM 快照的主要目的。LVM 的快照功能采用写时复制技术(Copy-On-Write, COW)，这比传统的备份技术的效率要高很多。创建快照时不用停止服务，就可以对数据进行备份。

LVM 采用的写时复制，是指当 LVM 快照创建的时候，仅创建到实际数据的 inode 的硬链接(hard-link)而已。只要实际的数据没有改变，快照就只包含指向数据的 inode 的指针，而非数据本身。快照会跟踪原始卷中块的改变，一旦你更改了快照对应的文件或目录，这个时候原始卷上将要改变的数据会在改变之前拷贝到快照预留的空间。

LVM快照技术

创建快照实际上也是创建了一个逻辑卷，只不过该卷的属性与普通逻辑卷的属性有些不一样。我们可以通过下图来理解快照数据卷(图中的实线框表示快照区域，虚线框表示文件系统)。



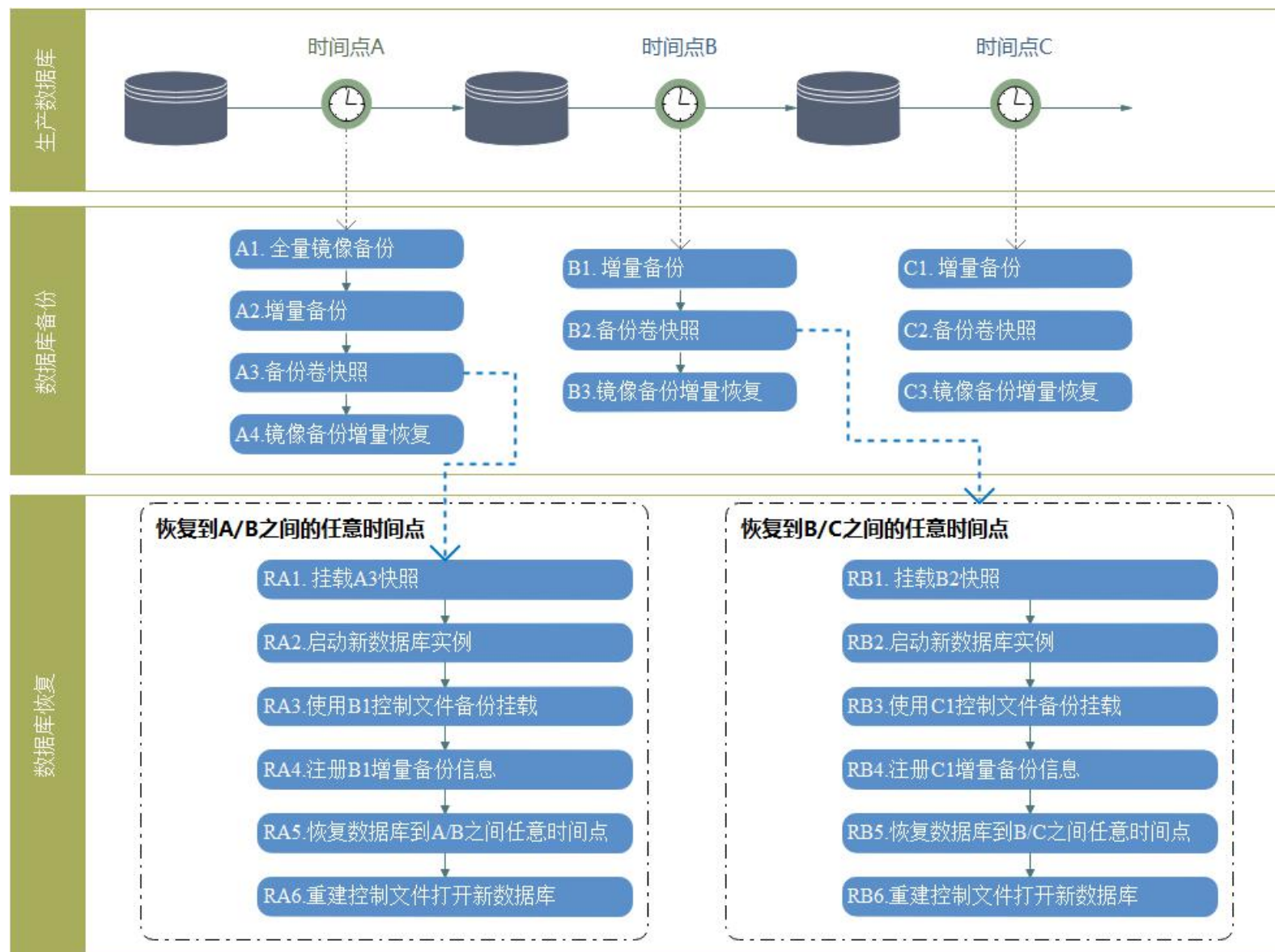
左图为最初创建的快照数据卷状况，LVM 会预留一个区域 (比如左图的左侧三个 PE 区块) 作为数据存放处。此时快照数据卷内并没有任何数据，而快照数据卷与源数据卷共享所有的 PE 数据，因此你会看到快照数据卷的内容与源数据卷中的内容是一模一样的。等到系统运行一阵子后，假设 A 区域的数据被更新了(上面右图所示)，则更新前系统会将该区域的数据移动到快照数据卷中，所以在右图的快照数据卷中被占用了一块 PE 成为 A，而其他 B 到 I 的区块则还是与源数据卷共享。

05

快速恢复



快速恢复



快速恢复

0. 配置恢复环境

口令文件

参数文件

挂载快照

1. 重建控制文件

```
SQL> alter database backup controlfile to trace;
```

```
SQL> CREATE CONTROLFILE REUSE DATABASE "SRCDB"  
NORESETLOGS.....;
```

```
SQL> SELECT NAME FROM V$DATAFILE;
```


快速恢复

2. 检查需要注册的归档日志文件

```
RMAN> list backup of archivelog all;
```

```
SQL> ALTER DATABASE REGISTER LOGFILE  
'/oracle/archive/1_11_961988430.dbf';
```

3. 恢复并打开数据库

```
SQL> RECOVER DATABASE;
```

```
SQL> ALTER DATABASE OPEN RESETLOGS;
```

```
SQL> ALTER TABLESPACE TEMP ADD TEMPFILE  
'/ora_backup_snap/ora11g_image/ORA11G-TEMP-1.dbf' SIZE 10M;
```


延迟恢复

```
RUN {  
    RECOVER COPY OF DATABASE WITH TAG  
'mydb_incr_backup' UNTIL TIME 'SYSDATE - 7';  
    BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY  
WITH TAG 'mydb_incr_backup' DATABASE;  
}
```

快速恢复-优势

✓ 传统方案创建测试库

独立硬件设备，存储空间

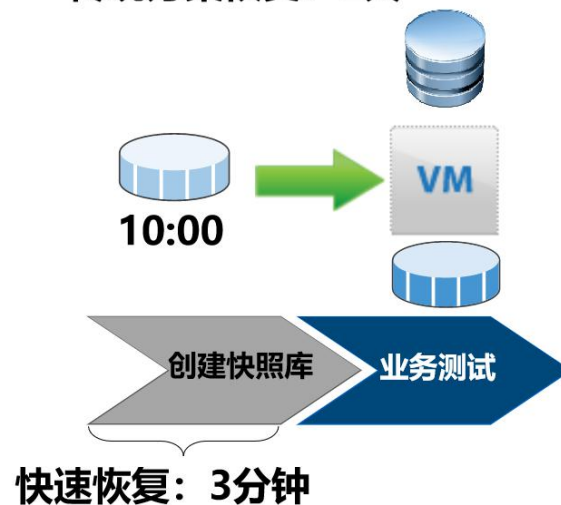
手工全库恢复



✓ 快速备份方案

分钟级创建

数据复用，无需拷贝



快速恢复-优势

1. 占用空间小，只需要一份全量镜像备份数据，后续的增量备份，在全量镜像备份基础上进行恢复。
2. 数据库恢复过程中不需要进行还原操作，只需要将最近一次增量备份恢复到全量镜像备份上，重建数据库控制文件，即可打开新的数据库。
3. 通过卷快照技术，可以方便的将数据库恢复到第一次全量镜像备份后的任意时间点，所需要恢复的数据是从上一次全量镜像恢复的时间点，到需要恢复时间点之间的数据变化，恢复所需的时间极为短暂。
4. 使用快照技术后，数据库恢复后占用的空间容量很小，尤其是在数据库只做查询的情况下，只需要整库容量的20%左右，在启动多个数据库进行数据检查时，效果尤为明显。

快速恢复-优势

5. 备份数据可以根据需要，**随时启动**，用于代码测试，功能验证，数据恢复等工作，不需要漫长的还原过程。传统的数据库备份只能用于整库恢复，并且需要先进行还原，在恢复到指定时间点。
6. 目前所有操作系统都支持原生操作系统卷快照技术和第三方卷管理软件，因此**该方式适用于基本所有的操作系统平台**。
7. 该方法使用了操作系统卷快照技术和ORACLE数据库的RMAN备份技术，这些都是系统或者软件自带的功能，**无需引入其他第三方软件**，不会产生额外的费用和成本。
8. 通过NFS或者虚拟化技术，快照卷可以挂载到新的主机或虚拟机上，**扩展性强**，性能优势大。

谢谢！