一、检查点概述  
大多数关系型数据库都采用"在提交时并不强迫针对数据块的修改完成"而是"提交时保证修改记录（以重做日志的形式）写入日志文件"的机制，来获得性能的优势。这句话的另外一种描述是：**当用户提交事务，写数据文件是"异步"的，写日志文件是"同步"的**。这就可能导致数据库实例崩溃时，内存中的DB\_Buffer 中的修改过的数据，可能没有写入到数据块中。数据库在重新打开时，需要进行恢复，来恢复DB Buffer 中的数据状态，并确保已经提交的数据被写入到数据块中。**检查点是这个过程中的重要机制，通过它来确定，恢复时哪些重做日志应该被扫描并应用于恢复**。  
要了解这个检查点，首先要知道checkpoint queue概念，检查点**发生后**，触发dbwr，**CKPT获取发生检查点时对应的SCN，通知DBWr要写到这个SCN为止**，  
dbwr 写dirty buffer 是根据 buffer 在**被首次修改**的时候的时间的顺序写出，也就是 buffer被修改的时候会进入一个queue （checkpoint queue），**dbwr 就根据queue从其中批量地写到数据文件**。 由于这里有一个顺序的关系，所以 dbwr的写的进度就是可衡量的，写到哪个buffer的时候该buffer的首次变化时候的scn就是当前所有数据文件block的最新scn，但是由于无法适时的将dbwr的进度记录下来，所以oracle 选择了一些策略。 其中就包括ckpt进程的检查点和心跳。  
检查点发生以后，CKPT进程检查checkpoint queue(也就是脏块链表)是否过长，如果是，则触发DBWr，将一部分脏块写入数据文件，从而缩短checkpoint queue。

checkpoint 发生时，一方面通知dbwr进行下一批写操作，（dbwr 写入的时候，一次写的块数是有一个批量写的隐藏参数控制的。）另一方面，oracle 采用了一个心跳的概念，**以3秒的频率将dbwr 写的进度反应到控制文件中**，也就是把dbwr当前刚写完的dirty buffer对应的scn 写入数据文件头和控制文件，这就是检查点scn。  
这个3秒和增量检查点不是一个概念，3秒只是**在控制文件中，ckpt 进程去更新当前dbwr写到哪里了**，这个对于ckpt 进程来说叫 heartbeat ，heartbeat是3秒一次，3秒可以看作不停的检查并记录检查点执行情况（DBWR的写进度）。  
检查点发生之后数据库的数据文件、控制文件处于一致状态的含义是不需要进行介质恢复，只表示数据文件头一致，但是并不表示数据文件内容一致，因为数据文件内容可能在没有发生检查点的其他情况下的dbwr写数据文件，这样数据文件内容就不一致，若掉电需要进行崩溃恢复。

二、SCN

SCN（System Chang Number）作为oracle中的一个重要机制，在数据恢复、Data Guard、Streams复制、RAC节点间的同步等各个功能中起着重要作用。理解SCN的运作机制，可以帮助你更加深入地了解上述功能。

1、系统检查点scn   
当一个检查点动作完成后，Oracle就把系统检查点的SCN存储到控制文件中。  
select checkpoint\_change# from v$database  
2、数据文件检查点scn   
当一个检查点动作完成后，Oracle就把每个数据文件的scn单独存放在控制文件中。  
select name,checkpoint\_change# from v$datafile   
3、启动scn  
Oracle把这个检查点的scn存储在每个**数据文件的文件头中**，这个值称为**启动scn**，因为它**用于在数据库实例启动时，检查是否需要执行数据库恢复**。  
select name,checkpoint\_change# from v$datafile\_header  
4、终止scn  
每个数据文件的**终止scn都存储在控制文件中**。  
select name,last\_change# from v$datafile  
在正常的数据库操作过程中，所有正处于联机读写模式下的数据文件的终止scn都为null.  
5、在数据库运行期间的scn值  
**在数据库打开并运行之后，控制文件中的系统检查点、控制文件中的数据文件检查点scn和每个数据文件头中的启动scn都是相同的**。控制文件中的每个数据文件的终止scn**都为null**.  
**在安全关闭数据库的过程中，系统会执行一个检查点动作，这时所有数据文件的终止scn都会设置成数据文件头中的那个启动scn的值**。在数据库重新启动的时候，Oracle将文件头中的那个启动scn与数据库文件检查点scn进行比较，如果这两个值相互匹配，oracle接下来还要比较数据文件头中的启动 scn和控制文件中数据文件的终止scn。如果这两个值也一致，就意味着所有数据块多已经提交，所有对数据库的修改都没有在关闭数据库的过程中丢失，因此这次启动数据库的过程也不需要任何恢复操作，此时数据库就可以打开了。当所有的数据库都打开之后，存储在控制文件中的数据文件终止scn的值再次被更改为 null，这表示数据文件已经打开并能够正常使用了。

三、事务过程

我们再看下oracle事务中的数据变化是如何写入数据文件的：

1、 事务开始；

2、 在buffer cache中找到需要的数据块，如果没有找到，则从数据文件中载入buffer cache中；

3、 事务修改buffer cache的数据块，该数据被标识为“脏数据”，并被写入log buffer中；

4、 事务提交，LGWR进程将log buffer中的“脏数据”写入redo log file中；

5、 当发生checkpoint，CKPT进程更新所有数据文件的文件头中的信息，DBWr进程则负责将Buffer Cache中的脏数据写入到数据文件中。

经过上述5个步骤，事务中的数据变化最终被写入到数据文件中。**但是，一旦在上述中间环节时，数据库意外宕机了，在重新启动时如何知道哪些数据已经写入数据文件、哪些没有写呢**（同样，在DG、streams中也存在类似疑问：redo log中哪些是上一次同步已经复制过的数据、哪些没有）？SCN机制就能比较完善的解决上述问题。

**SCN是一个数字，确切的说是一个只会增加、不会减少的数字**。正是它这种只会增加的特性确保了Oracle知道哪些应该被恢复、哪些应该被复制。

总共有**4种**SCN：**系统检查点**（System Checkpoint）SCN、**数据文件检查点**（Datafile Checkpoint）SCN、**结束SCN**（Stop SCN）、**开始SCN**（Start SCN）。其中**前面3中SCN存在于控制文件中，最后一种则存在于数据文件的文件头中**。

在控制文件中，System Checkpoint SCN是针对整个数据库**全局的，因而只存在一个**，而Datafile Checkpoint SCN和Stop SCN是针对每个数据文件的，因而一个数据文件就对应在控制文件中存在一份Datafile Checkpoint SCN和Stop SCN。在数据库正常运行期间，Stop SCN(通过视图v$datafile的字段last\_change#可以查询)是一个无穷大的数字或者说是NULL。

在一个事务提交后（上述第四个步骤），会在redo log中存在一条redo记录，同时，系统为其提供一个最新的SCN（通过函数dbms\_flashback.get\_system\_change\_number可以知道当前的最新SCN），记录在该条记录中。如果该条记录是在redo log被清空（日志满做切换时或发生checkpoint时，所有变化日志已经被写入数据文件中），则其SCN被记录为redo log的**low SCN**。以后在日志再次被清空前写入的redo记录中SCN则成为**Next SCN**。

当日志切换或发生checkpoint（上述第五个步骤）时，从Low SCN到Next SCN之间的所有redo记录的数据就被DBWn进程写入数据文件中，而CKPT进程则将所有数据文件（无论redo log中的数据是否影响到该数据文件）的文件头上记录的Start SCN(通过视图v$datafile\_header的字段checkpoint\_change#可以查询)更新为Next SCN，同时将控制文件中的System Checkpoint SCN（通过视图v$database的字段checkpoint\_change#可以查询）、每个数据文件对应的Datafile Checkpoint（通过视图v$datafile的字段checkpoint\_change#可以查询）也更新为Next SCN。但是，如果该数据文件所在的表空间被设置为read-only时，数据文件的Start SCN和控制文件中Datafile Checkpoint SCN都不会被更新。

四、心跳

在Oracle中有一个事件叫Heartbeat，这个词在很多地方被提及，并且有着不同的含义(比如RAC中)，我们这里要讨论的是CKPT的Heartbeat机制。

Oracle通过CKPT进程每3秒将Heartbeat写入控制文件，以减少故障时的恢复时间(这个我们后面再详细阐述)。

我们可以通过如下方法验证这个过程。

1.首先在系统级启用10046时间跟踪

并重新启动数据库使之生效

[oracle@jumper oracle]$ sqlplus "/ as sysdba"  
SQL\*Plus: Release 9.2.0.4.0 - Production on Thu Jan 19 09:24:04 2006  
Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.  
Connected to:Oracle9i Enterprise Edition Release 9.2.0.4.0 - ProductionWith the Partitioning optionJServer Release 9.2.0.4.0 - Production  
SQL> alter system set event='10046 trace name context forever,level 12' scope=spfile;  
System altered.  
SQL> shutdown immediate;Database closed.Database dismounted.ORACLE instance shut down.  
SQL> startupORACLE instance started.  
Total System Global Area 114365800 bytesFixed Size                   451944 bytesVariable Size              50331648 bytesDatabase Buffers           62914560 bytesRedo Buffers                 667648 bytesDatabase mounted.Database opened.SQL> exitDisconnected from Oracle9i Enterprise Edition Release 9.2.0.4.0 -   
ProductionWith the Partitioning optionJServer Release 9.2.0.4.0 - Production   
2.检查bdump目录下生成的跟踪文件

目录在$ORACLE\_HOME/admin/$ORACLE\_SID/bdump目录下，每个后台进程都会生成一个跟踪文件。

[oracle@jumper bdump]$ ls20050424\_alert\_conner.log conner\_arc0\_2569.trc   
conner\_dbw0\_2559.trc conner\_reco\_2567.trcalert\_conner.log           conner\_arc1\_2571.trc   
conner\_lgwr\_2561.trc conner\_smon\_2565.trca.sql                      conner\_ckpt\_2563.trc   
conner\_pmon\_2557.trc  
3.检查CKPT进程的跟踪文件

我们可以很容易的发现CKPT进程每3秒都对控制文件进行一次写入

[oracle@jumper bdump]$ tail -f conner\_ckpt\_2563.trc WAIT #0: nam='rdbms ipc message' ela= 2994710 p1=300 p2=0 p3=0WAIT #0: nam='control file parallel write' ela= 2442 p1=3 p2=3 p3=3WAIT #0: nam='rdbms ipc message' ela= 2995171 p1=300 p2=0 p3=0WAIT #0: nam='control file parallel write' ela= 2586 p1=3 p2=3 p3=3WAIT #0: nam='rdbms ipc message' ela= 2994962 p1=300 p2=0 p3=0WAIT #0: nam='control file parallel write' ela= 2582 p1=3 p2=3 p3=3WAIT #0: nam='rdbms ipc message' ela= 2995020 p1=300 p2=0 p3=0WAIT #0: nam='control file parallel write' ela= 2455 p1=3 p2=3 p3=3WAIT #0: nam='rdbms ipc message' ela= 2995188 p1=300 p2=0 p3=0WAIT #0: nam='control file parallel write' ela= 2412 p1=3 p2=3 p3=3WAIT #0: nam='rdbms ipc message' ela= 2995187 p1=300 p2=0 p3=0WAIT #0: nam='control file parallel write' ela= 2463 p1=3 p2=3 p3=3WAIT #0: nam='rdbms ipc message' ela= 2995095 p1=300 p2=0 p3=0WAIT #0: nam='control file parallel write' ela= 2448 p1=3 p2=3 p3=3  
4.检查控制文件的变更

通过2次dump控制文件，比较其trace文件输出可以比较得到不同之处，我们发现，Oracle仅仅更新了Heartbeat这个数值。

[oracle@jumper udump]$ sqlplus "/ as sysdba"  
SQL\*Plus: Release 9.2.0.4.0 - Production on Wed Jan 18 22:44:10 2006  
Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.  
Connected to:Oracle9i Enterprise Edition Release 9.2.0.4.0 - ProductionWith the Partitioning optionJServer Release 9.2.0.4.0 - Production  
SQL> alter session set events 'immediate trace name CONTROLF level 10';  
Session altered.  
SQL> exitDisconnected from Oracle9i Enterprise Edition Release 9.2.0.4.0 - ProductionWith the Partitioning optionJServer Release 9.2.0.4.0 - Production[oracle@jumper udump]$ sqlplus "/ as sysdba"  
SQL\*Plus: Release 9.2.0.4.0 - Production on Wed Jan 18 22:44:18 2006  
Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.  
Connected to:Oracle9i Enterprise Edition Release 9.2.0.4.0 - ProductionWith the Partitioning optionJServer Release 9.2.0.4.0 - Production  
SQL> alter session set events 'immediate trace name CONTROLF level 10' ;  
Session altered.  
SQL> exitDisconnected from Oracle9i Enterprise Edition Release 9.2.0.4.0 - ProductionWith the Partitioning optionJServer Release 9.2.0.4.0 - Production  
[oracle@jumper udump]$ lsconner\_ora\_21896.trc conner\_ora\_21898.trc[oracle@jumper udump]$ diff conner\_ora\_21896.trc conner\_ora\_21898.trc 1c1< /opt/oracle/admin/conner/udump/conner\_ora\_21896.trc---> /opt/oracle/admin/conner/udump/conner\_ora\_21898.trc14c14< Unix process pid: 21896, image: oracle@jumper.hurray.com.cn (TNS V1-V3)---> Unix process pid: 21898, image: oracle@jumper.hurray.com.cn (TNS V1-V3)16c16< \*\*\* SESSION ID9.813) 2006-01-18 22:44:14.314---> \*\*\* SESSION ID9.815) 2006-01-18 22:44:21.569  
63c63< heartbeat: 579991793 mount id: 3191936000---> heartbeat: 579991796 mount id: 3191936000 [oracle@jumper udump]$