第 7 章 数据库体系结构

Oracle 和 SQL Server 都是关系型 DBMS 软件，在体系结构方面，有诸多相似之处。但两个产品脱胎于不同的源头(Oracle 源自 IBM 的 System R，SQL Server 源自 Sybase，而 Sybase 又源自 UCB 的 Ingres)，在细节方面也注定存在不少区别。

本章主要内容包括：

- 服务器结构
- 数据库文件及数据库相关文件
- 主要进程
- SQL Server 的系统数据库
- 客户端连接的处理模式

7.1 服务器结构

Oracle 的服务器(server)由实例(instance)及数据库(database)构成。实例包括 Oracle 占用的内存结构以及后台进程，数据库包括数据文件、重做日志文件以及控制文件三种文件。

与 Oracle 类似，SQL Server 服务器也可以看作是由实例及数据库构成。实例包括 SQL Server 占用的内存及后台线程。与 Oracle 显著不同的是，SQL Server 服务器上的数据库是多个，其中包括 5 个系统数据库(resource 系统数据库对用户不可见)及若干个用户数据库，而 Oracle 服务器和数据库是合而为一的。

Oracle 数据库文件包括控制文件、数据文件及重做日志文件，SQL Server 数据库文件只包含数据文件和重做日志文件。

7.2 数据库文件及数据库相关文件

数据库文件是指数据库正常运行必不可少的文件，数据库相关文件是指数据库运行会用到，但并不是必不可少，或者在某些时刻并不是必不可少的文件。

7.2.1 Oracle 的情形

Oracle 的数据库文件包括三种：

- 数据文件
- 重做日志文件
- 控制文件

数据文件用于存放数据库中的数据。

重做文件存放用户对数据库的操作记录，用于实例恢复或介质恢复。Oracle 数据库正常运行最少要有两组重做文件，每组中的重做文件是镜像关系，同组中的重做文件大小相同。一组重做文件写满后，会切换到另一组。重做文件以 squence 编号及重做文件中的 SCN 号范围来标识(low SCN 及 next SCN，low SCN 是一个重做文件中的最小的 SCN 号，next SCN 是下一个重做文件中的最小的 SCN 号)，sequence 编号由 1 开始，在日志切换的过程中每次加 1，逐渐增大。如果配置了数据库的归档模式，则在切换到另一组的同时，会把之前正在写入的重做文件归档到指定目录(即拷贝重做文件中的重做数据)。

控制文件包含了数据库中的数据文件与重做文件的信息，除此之外还保存了表空间信息、重做文件的历史信息以及 rman 备份信息等数据。

在启动数据库时，Oracle 读取参数文件启动实例并得知控制文件的路径(此步骤称为 nomount)，再读取控制文件得到数据文件及重做文件的路径(此步骤称为 mount)，最后通过这些信息打开数据文件及重做文件，数据库就正常可用了(此步骤称为 open)。

与数据库相关的其他文件还包括：

- 初始化参数文件
- 口令文件
- 归档日志文件
- 警告文件

初始化参数文件用于保存实例启动及运行时的各种参数配置，实例启动时，初始化参数文件是必须的。初始化参数文件的默认位置为：%ORACLE_HOME%\database，这里的 ORACLE_HOME 表示安装 Oracle 软件的目录的环境变量，文件名称为：spfile*sid*.ora，sid 为实例名称。初始化参数分为动态参数与静态参数，动态参数可以使用命令修改并立即生效，而不必重启数据库，修改静态参数则要通过修改初始化参数文件实现，重启数据库才能使其生效。

口令文件保存 sys 用户及具备 sysdba 系统权限的用户的口令。Oracle 中的用户及其口令一般都存储在数据库中，但是 sys 用户例外，sys 用户除了在数据库中拥有管理权限外，还拥有启动和关闭数据库等特殊权限，如果 sys 用户的口令也与其他用户的口令一样存储在数据库中，在数据库打开之前无法验证其口令的正确性。

除了 sys 用户的口令外，口令文件还存储了其他被授予 sysdba 系统权限的用户的名称及口令。口令文件所在的目录一般为：%ORACLE_HOME%\database，其名称为 pwd*sid*.ora。

如果数据库运行在归档模式下，重做文件发生日志切换时，这个重做日志文件会同时被归档，即把文件中的重做数据拷贝到指定目录，以用于数据库恢复。

警告文件是一个简单的文本文件，可以看作数据库运行情况的记录，从数据库创建开始一直到被删除，数据库运行的信息都会被记录在这个文件中。通过这个文件，可以知道什么时候日志发生了切换(log switch)，什么时候发生了内部错误，什么时候创建了表空间，什么时候把表空间或数据文件脱机、联机，什么时候数据库被关闭、启动等等信息。出现错误时，若不能确定原因，应首先查看警告文件的内容，得到解决问题的线索。若管理 Oracle 软件的 Windows 用户为 oracle，安装在 C 分区，则警告文件位于 C:\app\oracle\diag\rdbms\law\law\trace 目录下，名称为 alert_*sid*.log。

7.2.2 SQL Server 的情形

SQL Server 的数据库文件包括：

- 数据文件
- 重做日志文件

数据文件与重做日志文件的作用与 Oracle 的对应文件相同，只是 SQL Server 的重做日志文件除了包含重做数据外，还包含回滚事务所用的 undo 数据，Oracle 的重做日志文件只包含重做数据，undo 数据存储在 undo 表空间。

SQL Server 没有控制文件，实例中的各个数据库文件的信息存储在 master 系统数据库以及用户数据库的 primary 文件组的主数据文件中。

SQL Server 没有初始化参数文件，实例的配置信息保存在 master 系统数据库中，数据库的配置信息保存在各自数据库的 primary 文件组的主数据文件中。

SQL Server 没有口令文件，启动 SQL Server 各种服务由操作系统帐号完成，其口令由

操作系统维护。

SQL Server 没有归档日志文件，Oracle 归档日志的功能通过事务日志文件备份实现。

Oracle 数据库的警告文件在 SQL Server 中称为错误日志(Errorlog)，是实例范围的，而不是针对某个数据库。与 Oracle 的警告文件类似，由 SQL Server 错误日志是文本文件，可以用来查看实例启动过程，以及实例运行过程中出现的错误或潜在的问题。

若 SQL Server 安装在 D 分区，SQL Server 的错误日志文件的位置为：

D:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\Log

服务器启动时，会创建新的错误日志文件 ERRORLOG，上一次的 ERRORLOG 被重命名为 ERRORLOG.1，ERRORLOG.1 被重命名为 ERRORLOG.2，依此类推，一直到 ERRORLOG.5 被重命名为 ERRORLOG.6，而 ERRORLOG.6 被删除，这样，错误日志最多保留 6 个备份。执行 sp_cycle_errorlog 系统存储过程可以自动创建新的 ERRORLOG 文件并执行上述修改文件名称的过程，而不必重启服务器。

可以使用任何文本编辑器在操作系统中查看其内容，也可以在 Management Studio 中通过“管理”→“SQL Server 日志”查看其内容。下面图示在 Management Studio 中打开当前错误日志：

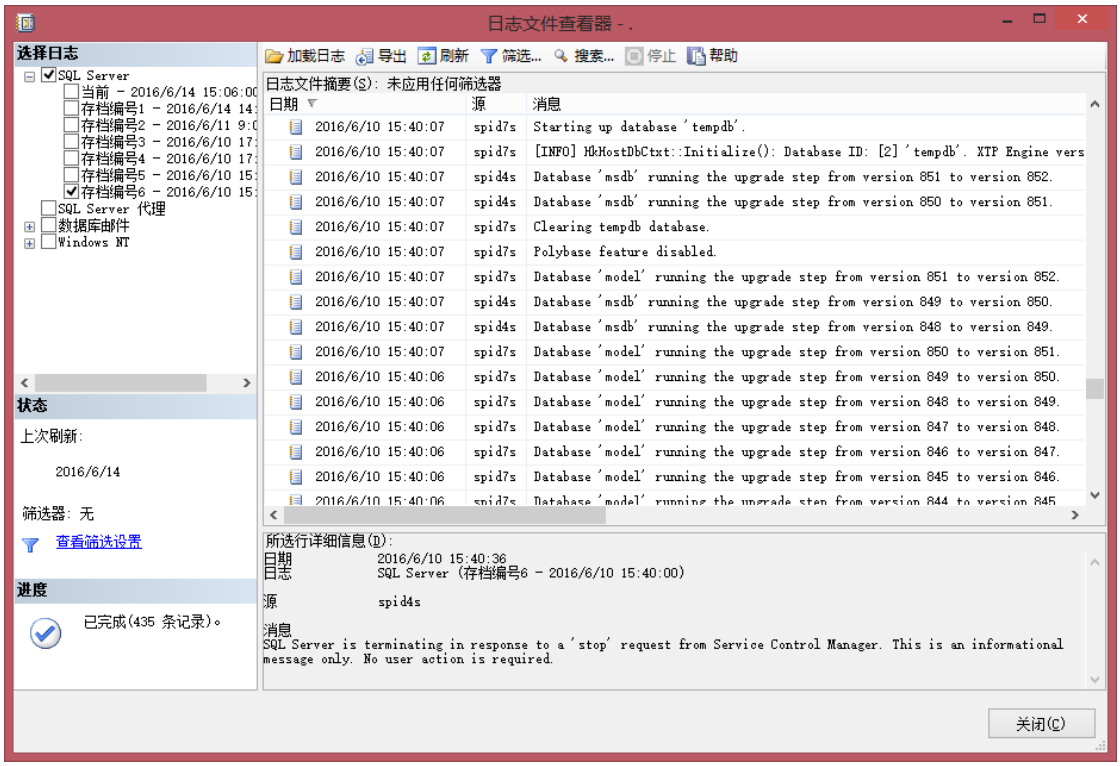

图 7-1 查看SQL Server日志文件

7.3 内存结构

内存结构是数据库服务器实例的重要部分，随着版本的不断升级，Oracle 数据库内存结构需要人工调整的任务越来越少，这与 SQL Server 逐渐接近，一般只需要根据环境需要和物理内存大小，指定分配给数据库服务器的总内存大小即可。

7.3.1 Oracle 的内存结构

Oracle 的内存结构主要分为 SGA(System Global Area)及 PGA(Process Global Area)，前者为 Oracle 所有进程共享的内存区域，后者为各个进程私有的内存区域。

SGA 是所有的 Oracle 进程可以访问的一个内存区域。在 UNIX 系统上，这块内存由一个共享内存段(shared memory segment)实现，Oracle 进程可以连接到(attach)这块区域访问其数据。在 Windows 系统，SGA 是分配给进程 oracle.exe 的内存空间，在 UNIX 上的各独立进程在 Windows 上作为 oracle.exe 的线程存在，这些线程共享 oracle.exe 的内存空间。

下面命令是使用 ipcs 命令在 Linux 系统查看共享内存的情况，其第二行即表示分配给 Oracle 实例的 SGA，其大小为 171966464 字节，当前有 19 个进程访问这块内存：

```
[root@law kernel]# ipcs -m
------ Shared Memory Segments --------
key          shmid      owner      perms      bytes      nattch     status
0x00000000   32768      gdm         600       393216     2          dest
0xde684ed0   65537      oracle      600       171966464  19
```

SGA 主要包括数据缓冲区，重做缓冲区，共享池，大池，java 池等区域。

数据缓冲区用于存放由磁盘读取的数据，目的是以后不必从磁盘再次读取。一般情况下，这是 SGA 中最大的一个区域。

当用户执行数据修改操作时(如 update 操作)，对这个客户端连接进行服务的服务器进程(server process)会先生成执行这个操作的重做记录(Redo record)，保存于重做缓冲区中，然后使用这个重做记录来执行对数据块的实际修改操作，在一定条件下，Redo log buffer 中的内容会由 LGWR 进程写入磁盘上的重做日志文件。

共享池分为 library cache 以及 dictionary cache 两部分，library cache 存放 SQL 语句的解析结果及执行计划，再次次执行这个语句，可以避免再次执行语法解析及重新生成执行计划，从而避免资源消耗。dictionary cache 存放被频繁访问的数据字典数据。

大池主要用于下面几种情况：共享服务器连接方式中，在 large pool 中分配 UGA；进程并发操作时(如并发查询)，保存进程间协调信息；rman 备份时，作为磁盘 I/O 缓冲区。

之所以称为大池，并不是因为这个区域很大，而是因为这个区域使用内存的方式特殊：SGA 区的其他部分，内存的使用方式主要是基于 LRU 算法，为了满足内存再次被访问的需要，需要空闲内存空间时，释放最久未访问的内存部分，而尽量保留最近被访问的内存中的数据，而上面列出的几种情况显然不是这样，而是需要时分配一大块内存，使用后完全释放供其他进程使用，一般不存在再次被访问的需要。

java 池用于支持在数据库中运行 JVM，当执行 Java 编写的存储过程时，会使用这个区域的内存。

对应于每个客户端连接都会有一个服务器进程处理其各种请求，这个服务器进程所占用的内存称为 PGA，这部分内存是只能由其对应的服务器进程访问的私有空间，其主要内容是排序区及散列区，用于在内存中完成排序或散列操作。

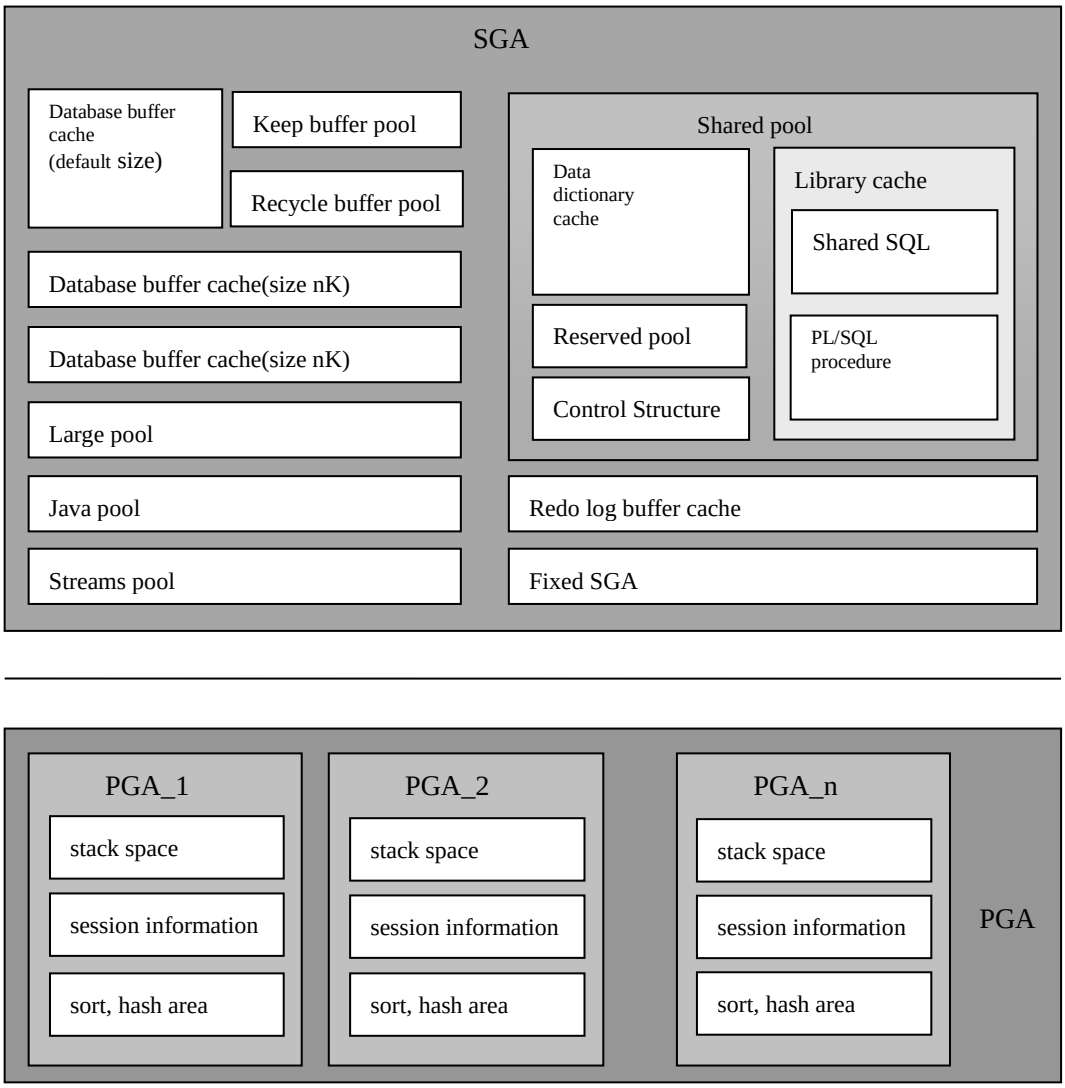

图 7-2 Oracle实例内存结构

查询内存各个部分的当前大小，可以使用下面命令：

```
SQL> select component, current_size
  2  from v$memory_dynamic_components
  3  /

COMPONENT                      CURRENT_SIZE
------------------------------ ------------
shared pool                       369098752
large pool                         33554432
java pool                          50331648
streams pool                              0
SGA Target                       1090519040
DEFAULT buffer cache              570425344
KEEP buffer cache                         0
RECYCLE buffer cache                      0
DEFAULT 2K buffer cache                   0
DEFAULT 4K buffer cache                   0
DEFAULT 8K buffer cache                   0
DEFAULT 16K buffer cache                  0
DEFAULT 32K buffer cache                  0
```

```
Shared IO Pool                 50331648
Data Transfer Cache                   0
PGA Target                     587202560
ASM Buffer Cache                      0

已选择 17 行。
```

用于配置 Oracle 内存大小的参数有两个：

- memory_target
- memory_max_target

memory_target 用于设置 Oracle 实例使用的内存总量，包括 SGA 以及所有的 PGA 总大小，SGA 中每个区域的大小以及 PGA 大小都由 Oracle 自动分配。

memory_max_target 用于指定 memory_target 参数可以设置的最大值。

memory_target 是动态初始化参数，可以使用下面命令直接修改：

```
SQL> alter system set memory_target=500m;
```

memory_max_target 是静态初始化参数，只能修改 spfile，然后重启数据库使其生效：

```
SQL> alter system set memory_max_target=800m scope=spfile;
```

7.3.2 SQL Server 的内存结构

SQL Server 的内存的主要由两部分构成：buffer cache 及其他部分。

buffer cache 也称为 buffer pool，是 SQL Server 内存的主要部分，其作用类似于 Oracle 的 SGA。buffer cache 中的主要部分为 data cache，相当于 Oracle 实例 SGA 中的 database buffer cache 部分。

buffer cache 中的另外一个重要部分为 plan cache，用于存放编译过的执行计划，相当于 Oracle 实例 shared pool 中的 library cache 部分。

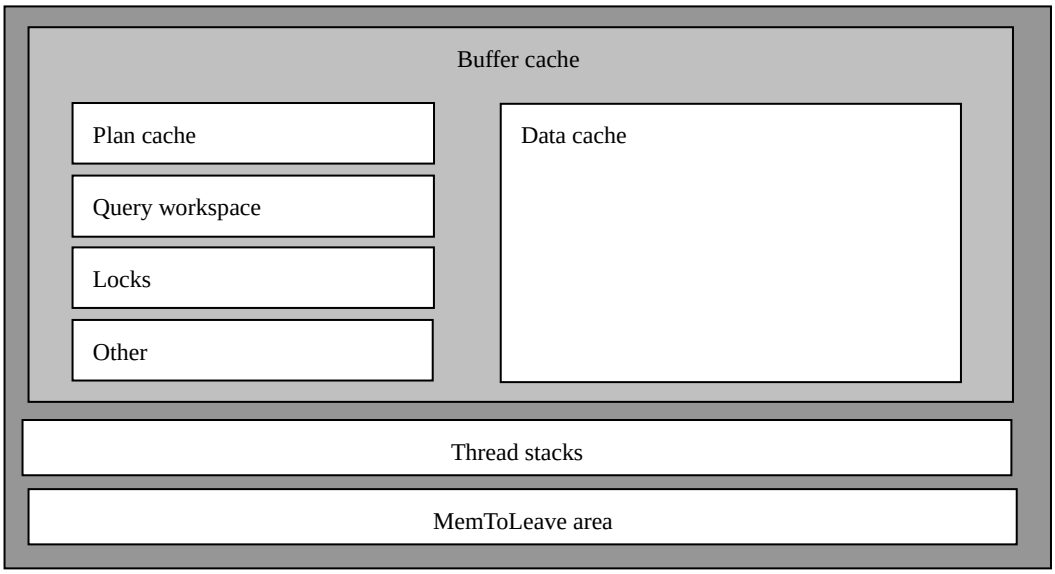

图 7-3 SQL Server 实例内存结构

与 SQL Server 内存分配相关的服务器参数有两个：

- max server memory：设置 buffer cache 的上限。
- min server memory：设置 SQL Server 可以释放内存的 buffer cache 下限。

min server memory 的默认设置为 0，max server memory 的默认设置为 2147483647。可以为 max server memory 指定的最小内存量为 16 MB，使用默认设置，即允许 SQL Server

根据系统资源的情况动态调整内存占用量。

下面是设置这两个参数的方法。

配置 max server memory 的值为 500MB：

```
1> sp_configure 'max server memory', '500'
2> go
配置选项 'max server memory (MB)' 已从 3000 更改为 500。请运行 RECONFIGURE 语句进行安装。
1> reconfigure
2> go
```

配置 min server memory 为 300MB：

```
1> sp_configure 'min server memory', '300'
2> go
配置选项 'min server memory (MB)' 已从 200 更改为 300。请运行 RECONFIGURE 语句进行安装。
1> reconfigure
2> go
```

一般情况下，SQL Server 的内存分配不需要用户干预，SQL Server 尽力做到获得尽量多的内存，又不会使系统出现内存短缺现象。

SQL Server 在启动时，根据当前负荷分配必要的内存数量，这个数量可能小于 min server memory 的值，如果负荷一直不大，其内存占用可能在很长时间内不会达到 min server memory 的值。

运行过程中，SQL Server 会随着负荷及用户连接数的增长继续分配内存，直到内存总量达到 max server memory 值，或者操作系统提示内存短缺为止。

当 SQL Server 占用的内存超过 min server memory 值，而且 Windows 系统因为其他应用的运行提示空闲内存缺少时，SQL Server 会释放内存，但会保持内存不低于 min server memory 的值，当这些应用退出时，SQL Server 又会获取更多的内存。在一秒钟之内，SQL Server 可以释放或获取几兆的内存。

如果 SQL Server 占用的内存尚未达到 min server memory 值，则这些内存会一直保持，而不会释放。

如果 max server memory 与 min server memory 配置为相同的值，内存占用量达到这个值后，不会继续分配也不会释放，这种方式可以使 SQL Server 占用固定数量的内存。

另外要注意，SQL Server 占用的内存总量可能会超过 max server memory 值，因为 max server memory 只是设置的 buffer cache 的上限，除此之外，SQL Server 还需要分配其他功能的内存。

7.4 主要进程(线程)

服务器实例的另外一部分是进程，执行数据写入及读取等各种任务。在 Windows 系统，Oracle 的各种进程都是以 oracle.exe 的子线程形式存在，在下面的叙述中，还是依照习惯称为进程，而 SQL Server 的进程以 sqlservr.exe 的子线程形式存在。

7.4.1 Oracle 的主要进程

Oracle 实例中的主要进程包括：DBWn，LGWR，SMON，PMON，CHECKPOINT。

DBWn 利用 LRU 算法把缓冲区中最少访问的脏块写入磁盘的数据文件，以增加数据缓冲区空闲空间大小。最多可以启动 20 个 DBWn 进程，DBW0 到 DBW9 以及 DBWa 到 DBWj，DBWn 进程的数量由初始化参数 DB_WRITER_PROCESSES 设定。在 DBWn 把脏块写入磁

盘之前，先由 LGWR 把 Redo log buffer 中的内容写入磁盘上的重做日志文件。每隔 3 秒钟或者 checkpoint 进程启动时，都会激发 DBWR 的执行。

LGWR 负责把 Log buffer 中的内容写入重做日志文件。事务提交完成的标志是在 Log buffer 中把与其相关的信息以及 commit 标记写入了重做文件，DBWn 进程把脏块写入数据文件时，首先要由 LGWR 把与这些脏块内的数据相关的重做信息写入重做文件后才能进行。LGWR 启动的条件包括用户发出了 commit 命令，Redo log buffer 达到了 1MB 或 1/3 满，每隔 3 秒钟等。

SMON 在数据库崩溃重启时，执行数据库恢复任务，另外还负责释放临时段，临时段是在内存不足时，临时表空间中用来存放排序或散列操作中间结果的磁盘空间。

PMON 当用户的连接异常断开时，回滚这个连接开始的事务，并释放与这个连接相关的资源，如锁、内存等。PMON 也给监听器进程提供连接请求的信息。

CHECKPOINT 启动时，会在重做文件中加入一个标志(SCN 号)，并启动 DBWn 进程把数据缓冲区中的脏块写入磁盘，这个 SCN 标志可以看作重做日志文件中的一个时间点，在此之前的脏块都已经被写入磁盘，这样 checkpoint 可以用来作为实例恢复的开始点，另外 Oracle 的 checkpoint 进程启动时，还会遍历数据文件及控制文件，在其文件头上写入当前的 SCN 号作为同步信息。重做日志切换，表空间脱机、表空间置于热备份或只读状态等操作都会激发 checkpoint 启动，另外，也可以手工执行 checkpoint 命令，把脏块写入磁盘。

7.4.2 SQL Server 的主要线程

SQL Server 与 Oracle 的 DBWn 和 checkpoint 进程对应的是 lazy writer 和 checkpoint。

lazy writer 运行的目的是增加 data cache 空闲内存，并保持一定的系统空闲内存。当 data cache 中的空闲内存不够时，lazy writer 搜索 data cache，把脏块写入磁盘，并把这些可以重用的内存页放入自由列表(free list)，以增大空间内存数量。另外，lazy writer 会缩小或扩充 data cache 的大小，使得系统空闲内存保持在 5MB 左右。

checkpoint 的目的是缩短数据库恢复时间。checkpoint 启动时，会搜索整个 data cache，把修改过的数据页写入磁盘的数据文件，从而保证内存中的脏数据块不会很多，当数据库发生崩溃再次重启时，checkpoint 会作为数据库恢复的起始点，从而重做(redo，或称为前滚)的时间不会过长，这与 Oracle 相同，但 checkpoint 把脏数据页写入磁盘后，并不把这些可以再次使用的内存数据页放入自由列表。与 Oracle 不同，SQL Server 的 checkpoint 并不会起到同步各种文件的作用。

如下情况都可以激发 checkpoint 启动：

- 用户发出 checkpoint 命令。
- 对数据库添加或删除了文件。
- 对大容量日志恢复模式的数据库执行了大容量操作(大容量操作请参考第 9 章内容)。
- 数据库处于简单恢复模式时，若重做文件中的数据量超过文件总大小的 70%，会启动 checkpoint 把脏块写入磁盘，checkpoint 会同时截断重做日志，以释放空间。若重做日志文件的充满是由于一个事务长时间未结束，则 checkpoint 不会启动。
- 预测恢复时间超过预设的 recovery interval 值，会启动 checkpoint。recovery interval 默认为 0，这时 SQL Server 自动选取一个合适值，一般为 1 分钟。
- 对数据库执行了备份操作。
- 正常关闭 SQL Server 实例服务。

lazy writer 与 checkpoint 都会把脏块写入磁盘，其主要区别是 checkpoint 并不会把这些可以重用的内存页放入自由列表。另外还要注意，并不只是 lazy writer 和 checkpoint 执行写磁盘操作，执行读写任务的 Work 线程(这里的 Work 线程相当于 Oracle 中对客户端连接提供

服务的服务器进程)在执行相关操作时，会检查 data cache 自由列表上的空闲内存是否过少，若过少，它也会把脏块写入磁盘，然后把这些内存页放入自由列表。checkpoint 启动时，可能无事可做，因为把脏块写入磁盘的任务已经被 lazy writer 或 Work 线程完成了。

7.5 SQL Server 的系统数据库

系统数据库包括 master、model、msdb、tempdb 以及 resource 数据库。

master：保存整个服务器的系统信息，如服务器配置信息，登录帐号信息，其他数据库的数据库文件信息等。

model：是数据库的模板，当用户创建新的数据库时，SQL Server 拷贝 model 数据库的结构作为新数据库的开始，用户可以修改这个数据库的选项设置，添加新用户或者创建各种数据库对象，以使其他新建的用户数据库都具备某些特征。用户不能对 model 数据库添加文件组，它只包含 primary 文件组，也不能向 primary 文件组添加新的数据文件，它只能包含一个主数据文件，但用户可以更改主数据文件和重做的大小及其他属性，如果在建库时未指定文件组及重做文件，则新数据库主数据文件会继承 model 数据库的主数据文件大小，但是其他如自动增长、最大大小等属性不会继承，新数据库的重做文件大小及属性也不会继承 model 数据库的重做文件的相应属性。

msdb：当配置了数据库的自动化管理时，msdb 数据库保存自动化作业的配置信息。

tempdb：类似 Oracle 数据库的临时表空间，用于保存临时表以及数据库运行过程中的排序或散列操作产生的临时数据。另外 tempdb 数据库还保存了用于实现行版本控制的数据(row version store)，这些数据的功能与 Oracle 数据库的 undo 表空间数据相似。

resource：保存 sys 架构的数据，主要是数据字典数据。在 Management Studio 中，这个数据库不会显示出来，用户也不能在 sqlcmd 中使用 use resource 命令登录这个数据库，而只能通过访问 sys 架构下的对象间接访问 resource 数据库中的内容。用户查询数据字典视图获得服务器或数数据库的系统信息，就是在访问这个数据库中的数据。

7.6 客户端连接的处理模式

Oracle 处理客户端连接包括专用服务器模式(dedicated server)和共享服务器模式(shared server)。

在专用服务器模式下，对应于每个客户端连接，在服务器端都会启动一个进程专门为其服务，这种进程称为服务器进程(Server process)，服务器进程处理客户端连接所提出的各种请求。当并发用户连接数量不是很大时(一般以 500 为限)，这是最常用的方式。

在共享服务器模式下，管理员可以手工设定并发服务器进程的数量，多个客户端连接会共用一个服务器进程。在这种模式下，还会启动另外的称为 Dispatcher 的进程，负责把服务器进程分配给客户端连接服务。

服务器进程占用的内存称为 PGA，主要用于其对应客户端连接的排序和散列操作。当并发连接数量很大时，一般并不是每个连接都在进行数据处理，这种情况下，专用服务器模式内存耗费过多，应采用共享服务器模式，以节省内存占用。

SQL Server 只有一种类似于 Oracle 的共享服务器模式的客户端连接处理模式。

对应于每个 CPU，SQL Server 会启动一个 Scheduler，可以看作是逻辑 CPU。Oracle 中处理客户端连接的服务器进程(Server Process)在 SQL Server 中称为 Work 线程(或纤程，取决于服务器配置)，每个 Work 线程大约占用 0.5MB 内存。

当有客户端请求时，会交给当前负荷最低的 Scheduler，如果这时没有空闲的 Work 线程，这个 Scheduler 会启动一个 Work 线程来处理这个请求，当一个 Work 线程在 15 分钟内都处于空闲状态，Scheduler 会销毁它以释放内存。