# Oracle数据库体系结构

参考<http://blog.csdn.net/robinson_0612/article/details/5529239>

## Oracle数据库体系结构

Oracle Server：一个Oracle Server由一个Oracle实例和一个Oracle数据库组成。即：Oracle Server = Oracle Instance + Oracle Database

Oracle Instance：包括了内存结构(SGA)和一系列后台进程(Background Process)。 即：Oracle Instance = SGA + Background Process

Oracle Memory Structures ：包含系统全局区(SGA)和程序全局区(PGA)。 即Oracle Memory Structures = SGA + PGA

SGA和PGA： SGA由服务器和后台进程共享。PGA包含单个服务器进程或单个后台进程的数据和控制信息，与几个进程共享的SGA 正相反，PGA是只被一个进程使用的区域，PGA 在创建进程时分配在终止进程时回收。即由服务器进程产生。

### 1.1、SGA

系统全局区SGA：SGA = 数据缓冲区+ 重做日志缓冲区+ 共享池+ 大池+ Java 池+ 流池

           系统全局区是动态的，由参数SGA＿MAX＿SIZE决定。

           查看当前系统的SGA大小：show parameter sga\_max\_size;

           要修改：alter system set sga\_max\_size=1200m scope=spfile;

           因为实例内存的分配是在数据库启动时进行的，所以要让修改生效，要重启数据库。

数据缓冲区(Database buffer cache)：存储从数据文件中获得的数据块的镜像

           大小由参数db\_cache\_size 决定。

           查看：show parameter db\_cache\_size;

           设置：alter system set db\_cache\_size=800M;

重做日志缓冲区(Redo log buffer)：对数据库的任何修改都按顺序被记录在该缓冲，然后由LGWR进程将 它写入磁盘,大小由LOG＿BUFFER决定。

共享池(Shared pool)：是SGA中最关键的内存片段,共享池主要由库缓存(共享SQL区和PL/SQL区)和数据字典缓存组成，它的作用是存放频繁使用的sql，在有限的容量下，数据库系统根据一定的算法决定何时释放共享池中的sql。

**库缓存**

大小由shared\_pool\_size 决定

              查看：show parameter shared\_pool\_size

              修改：alter system set shared\_pool\_size=120m;

**数据字典缓存：**

              存储数据库中数据文件、表、索引、列、用户和其它数据对象的定义和权限信息

              大小由shared\_pool\_size 决定，不能单独指定

大池(Large pool)：是一个可选的区域，用于一些大型的进程如Oracle的备份恢复操作、IO服务器进程等。

 Java 池：该程序缓冲区就是为Java 程序保留的。如果不用Java程序没有必要改变该缓冲区的默认大小。

流池(Stream pool)：被Oracle流所使用。

### 1.2、PGA

是为每个用户进程连接ORACLE数据库保留的内存。进程创建时分配，进程结束时释放，只能被一个进程使用。由参数：pga\_aggregate\_target 决定。

PGA包括了以下几个结构：

       （）排序区

       （）游标状态区

       （）会话信息区

       （）堆栈区

### 1.3、进程

用户进程：在用户连接数据库产生，请求oracle服务器连接，必须要先建立一个连接，不会直接和oracle服务器连接。

服务器进程： 当连接实例并建立用户会话时产生，独立服务器或者提供共享服务器都能产生。

后台进程：维持物理和内存之间的联系，用来管理数据库的读写，恢复和监视等工作。

           Server Process主要是通过他和user process进行联系和沟通，并由他和user process进行数据的交换。

           在Unix机器上，Oracle后台进程相对于操作系统进程，也就是说，一个Oracle后台进程将启动一个操作 系统进程。

           在Windows机器上，Oracle后台进程相对于操作系统线程，打开任务管理器，我们只能看到一个ORACLE.EXE的进程，但是通过另外的工具，就可以看到包含在这里进程中的线程。

必须要有的后台进程

       DBWn       -->数据库写进程

       PMON       -->程序监控进程

       SMON       -->系统监控进程

       LGWr       -->日志写进程

       CKPT       -->检查点进程

可选进程：

       ARCN       归档进程

       RECO

       Snnn

       pnnn

 DBWn(数据库写进程)

负责将修改过的数据块从数据库缓冲区高速缓存写入磁盘上的数据文件中，写入条件：

       发生检查点

       脏缓存达到限制

       没有自由的缓存

       超时发生

       表空间离线

       表空间只读

       表被删除或者截断

       开始备份表空间

           可以修改数据写进程的数量

           alter system set db\_writer\_processes=3 scope=spfile;

PMON(程序监控进程)

清除失效的用户进程，释放用户进程所用的资源。如PMON将回滚未提交的工作，释放锁，释放分配给失败进程的SGA资源。

       清除失败的进程

       回滚事务

       释放锁

       释放其他资源

SMON(系统监控进程)

检查数据库的一致性,当启动失败时完成灾难恢复等。实列恢复时，前滚所有重做日志中的文件，打开数据库为了用户能访问，回滚未提交的事务，释放临时表空间。清除临时空间，聚结空闲空间，从不可用的文件中恢复事务的活动，OPS中失败节点的实例恢复

       清除OBJ$表

       缩减回滚段

       使回滚段脱机

LGWr(日志写进程)

           将重做日志缓冲区中的更改写入在线重做日志文件。条件：

       提交的时候（commit)

       redo log buffer达到1／3满

       每隔3秒

       有大于1MB 重做日志缓冲区未被写入磁盘

       DBWR需要写入的数据的SCN号大于LGWR 记录的SCN号，DBWR 触发LGWR写入

       超时

       在dbwr进程些之前写日志

 CKPT(检查点进程)

           DBWR/LGWR的工作原理，造成了数据文件，日志文件，控制文件的不一致，CKPT进程负责同步数据文件，日志文件和控制文件， CKPT会更新数据文件/控制文件的头信息。条件：

       在日志切换的时候

       数据库用immediate ，transaction ，normal选项shutdown数据库的时候

       根据初始话文件LOG\_CHECKPOINT\_INTERVAL、LOG\_CHECKPOINT\_TIMEOUT、FAST\_START\_IO\_TARGET 的设置的数值来确定

       用户触发

 ARCN(归档进程)

           在每次日志切换时把已满的日志组进行备份或归档。条件：

       数据库以归档方式运行的时侯

RECO

           负责解决分布事物中的故障。Oracle可以连接远程的多个数据库，当由于网络问题，有些事物处于悬而未决的状态。

           RECO进程试图建立与远程服务器的通信，当故障消除后，RECO进程自动解决所有悬而未决的会话。

Server Process(服务进程)

           分为专用服务进程(Dedicated Server Process)和共享服务进程(MultiTreaded Server Process)

           专用服务进程：一个服务进程对应多个用户进程，轮流为用户进程服务。

用户进程(User Process)、服务进程(Server Process)、后台进程(Background Processes)的启动

           用户进程: 数据库用户请求Oralce server会话时被启动

           服务进程：当用户会话启动后，连接到Oracle实例时该进程被启动

           后台进程：当Oracle实例被启动时，启动相关的后台进程

## 2、Oracle 数据库

一系列物理文件的集合，包括控制文件、数据文件、联机日志文件、参数文件、密码文件等 。

即：Oracle Database = Controlfile + datafile + logfile + spfile +..

在ASM的硬盘下，数据库包括以下文件夹

ASMCMD> ls -l

Type Redund Striped Time Sys Name

Y CONTROLFILE/

Y DATAFILE/

Y ONLINELOG/

Y PARAMETERFILE/

Y TEMPFILE/

N spfilenms.ora => +RACDB\_DATA1/NMS/PARAMETERFILE/spfile.267.809348661

### 2.1、控制文件(CONTROLFILE)

数据库的名字，检查点信息，数据库创建的时间戳。

所有的数据文件，联机日志文件，归档日志文件信息，备份信息等

ASMCMD> pwd

+RACDB\_DATA1/NMS/CONTROLFILE

ASMCMD> ls -l

Type Redund Striped Time Sys Name

CONTROLFILE UNPROT FINE MAR 07 11:00:00 Y Current.256.809346543

ASMCMD> ls -s

Block\_Size Blocks Bytes Space Name

16384 1129 18497536 25165824 Current.256.809346543

### 2.2、数据文件(DATAFILE)

包含了用户和应用程序的所有数据 --查看数据文件信息

ASMCMD> pwd

+RACDB\_DATA1/NMS/DATAFILE

ASMCMD> ls -l

Type Redund Striped Time Sys Name

DATAFILE UNPROT COARSE MAR 07 11:00:00 Y SYSAUX.260.809346561

DATAFILE UNPROT COARSE MAR 07 11:00:00 Y SYSTEM.259.809346551

DATAFILE UNPROT COARSE MAR 07 11:00:00 Y UNDOTBS1.261.809346569

DATAFILE UNPROT COARSE MAR 07 11:00:00 Y UNDOTBS2.263.809346587

DATAFILE UNPROT COARSE MAR 07 11:00:00 Y USERS.264.809346591

ASMCMD> ls -s

Block\_Size Blocks Bytes Space Name

8192 76801 629153792 631242752 SYSAUX.260.809346561

8192 89601 734011392 736100352 SYSTEM.259.809346551

8192 96641 791683072 793772032 UNDOTBS1.261.809346569

8192 25601 209723392 211812352 UNDOTBS2.263.809346587

8192 641 5251072 6291456 USERS.264.809346591

### 2.3、联机日志文件(ONLINELOG)

记录了用户对数据库的所有操作，一个数据库中至少要有两个日志组文件，每个日志组中至少有一个日志成员，日志组中的多个日志成员是互为镜相关系。

ASMCMD> pwd

+RACDB\_DATA1/NMS/ONLINELOG

ASMCMD> ls -l

Type Redund Striped Time Sys Name

ONLINELOG UNPROT COARSE MAR 07 11:00:00 Y group\_1.257.809346547

ONLINELOG UNPROT COARSE MAR 07 11:00:00 Y group\_2.258.809346547

ONLINELOG UNPROT COARSE MAR 07 11:00:00 Y group\_3.265.809348657

ONLINELOG UNPROT COARSE MAR 07 11:00:00 Y group\_4.266.809348657

ASMCMD> ls -s

Block\_Size Blocks Bytes Space Name

512 102401 52429312 53477376 group\_1.257.809346547

512 102401 52429312 53477376 group\_2.258.809346547

512 102401 52429312 53477376 group\_3.265.809348657

512 102401 52429312 53477376 group\_4.266.809348657

### 2.4、归档日志文件

Oracle可以运行在两种模式之中，归档模式和非归档模式。在归档模式中，为了保存用户的所有修改，在联机日志文件切换后和被覆盖之间系统将他们另外保存成一组连续的文件系列，该文件系列就是归档日志文件。用户恢复意外情况出现的数据丢失、异常等。

ASMCMD> pwd

+FRA/NMS

ASMCMD> ls -l

Type Redund Striped Time Sys Name

Y CONTROLFILE/

Y ONLINELOG/

### 2.5、参数文件(pfile和spfile)

        initSID.ora或init.ora文件,通常位于：$ORACLE\_BASE/admin/<SID>/pfile

        初始化文件记载了许多数据库的启动参数，如内存，控制文件，进程数等，在数据库启动的时候加载(Nomount时加载)

ASMCMD> pwd

+Racdb\_data1/nms

ASMCMD> ls -s PARAMETERFILE/

Block\_Size Blocks Bytes Space Name

512 7 3584 1048576 spfile.267.809348661

ASMCMD> ls -s

Block\_Size Blocks Bytes Space Name

CONTROLFILE/

DATAFILE/

ONLINELOG/

PARAMETERFILE/

TEMPFILE/

spfilenms.ora => +RACDB\_DATA1/NMS/PARAMETERFILE/spfile.267.809348661

### 2.6、其他文件

密码文件：用于Oracle 的具有sysdba权限用户的认证。

告警日志文件：报警日志文件(alert.log或alrt.ora），记录数据库启动，关闭和一些重要的出错信息。

查看路径：select value from v$PARAMETER where name =‘background\_dump\_dest’;

### 2.7、数据库逻辑组织结构

表空间、段、区、块

一个数据库由一个或多个表空间组成，一个表空间只能属于一个数据库。

一个表空间由一个或多个多个数据文件组成，一个数据文件只能属于一个表空间。

一个数据文件由一个或多个操作系统块组成，每一个操作系统块只能数以一个数据文件。

一个表空间可以包含一个或多个段，一个段只能属于一个表空间。

一个段由一个或多个区组成，每一个区只能属于一个段。

一个区由一个或多个Oracle 块组成，每一个Oracle块只能属于一个区。

一个区只能属于一个数据文件，数据文件的空间可以分配到一个或多个区。

一个Oracle 块由一个或多个操作系统块组成，一个操作系统块是一个Oracle块的一部分。

## 3、Oracle实例和Oracle数据库的关系

1、一个实例能够装载及打开仅仅一个数据库

2、一个数据库能够被多个实例装载并打开

3、实例与数据库的对应关系是一对一或多对一的关系

## 4、数据库的概念

### 4.1、数据库用户、权限、角色概念

参考<http://blog.csdn.net/junmail/article/details/4381287>

SQL> desc dba\_users;

Name Null? Type

----------------------------------------- -------- ----------------------------

USERNAME NOT NULL VARCHAR2(30)

USER\_ID NOT NULL NUMBER

PASSWORD VARCHAR2(30)

ACCOUNT\_STATUS NOT NULL VARCHAR2(32)

LOCK\_DATE DATE

EXPIRY\_DATE DATE

DEFAULT\_TABLESPACE NOT NULL VARCHAR2(30)

TEMPORARY\_TABLESPACE NOT NULL VARCHAR2(30)

CREATED NOT NULL DATE

PROFILE NOT NULL VARCHAR2(30)

INITIAL\_RSRC\_CONSUMER\_GROUP VARCHAR2(30)

EXTERNAL\_NAME VARCHAR2(4000)

PASSWORD\_VERSIONS VARCHAR2(8)

EDITIONS\_ENABLED VARCHAR2(1)

AUTHENTICATION\_TYPE VARCHAR2(8)

### 4.2、数据库存储结构中的概念

参考http://space.itpub.net/17203031/viewspace-682003

#### ***数据块Block***

Block是Oracle存储数据信息的最小单位。查看block的大小，一般为8KB。

SQL> show parameter db\_block\_size;

NAME TYPE VALUE

------------------------------------ ---------------------------------------------------

db\_block\_size integer 8192

#### 区Extent

Extent是比Block大一级的存储结构，表示的是一连串连续的数据块集合。

和Extent相关的表（视图）dba\_extents。

SQL> desc dba\_extents;

Name Type Nullable Default Comments

--------------- ------------ -------- ------- ------------------------------------------------

OWNER VARCHAR2(30) Y Owner of the segment associated with the extent

SEGMENT\_NAME VARCHAR2(81) Y Name of the segment associated with the extent

PARTITION\_NAME VARCHAR2(30) Y Partition/Subpartition Name, if any, of the segment

SEGMENT\_TYPE VARCHAR2(18) Y Type of the segment

TABLESPACE\_NAME VARCHAR2(30) Y Name of the tablespace containing the extent

EXTENT\_ID NUMBER Y Extent number in the segment

FILE\_ID NUMBER Y Name of the file containing the extent

BLOCK\_ID NUMBER Y Starting block number of the extent

BYTES NUMBER Y Size of the extent in bytes

BLOCKS NUMBER Y Size of the extent in ORACLE blocks

RELATIVE\_FNO NUMBER Y Relative number of the file containing the segment header

#### 段Segment

段（segment）由一组区（extent）构成。数据段与数据库对象相对应，一般一个数据库对象对应一个数据段。多个extent对应一个数据段，每个数据段实际上就是数据库一个对象的代表。

SQL> desc dba\_segments;

Name Type Nullable Default Comments

--------------- ------------ -------- -------

OWNER VARCHAR2(30) Y Username of the segment owner

SEGMENT\_NAME VARCHAR2(81) Y Name, if any, of the segment

PARTITION\_NAME VARCHAR2(30) Y Partition/Subpartition Name, if any, of the segment

SEGMENT\_TYPE VARCHAR2(18) Y Type of segment: "TABLE", "CLUSTER", "INDEX", "ROLLBACK",

"DEFERRED ROLLBACK", "TEMPORARY","SPACE HEADER", "TYPE2 UNDO" or "CACHE"

TABLESPACE\_NAME VARCHAR2(30) Y Name of the tablespace containing the segment

HEADER\_FILE NUMBER Y ID of the file containing the segment header

HEADER\_BLOCK NUMBER Y ID of the block containing the segment header

BYTES NUMBER Y Size, in bytes, of the segment

BLOCKS NUMBER Y Size, in Oracle blocks, of the segment

EXTENTS NUMBER Y Number of extents allocated to the segment

INITIAL\_EXTENT NUMBER Y Size, in bytes, of the initial extent of the segment

NEXT\_EXTENT NUMBER Y Size, in bytes, of the next extent to be allocated to the segment

MIN\_EXTENTS NUMBER Y Minimum number of extents allowed in the segment

MAX\_EXTENTS NUMBER Y Maximum number of extents allowed in the segment

PCT\_INCREASE NUMBER Y Percent by which to increase the size of the next extent to be allocated

FREELISTS NUMBER Y Number of process freelists allocated in this segment

FREELIST\_GROUPS NUMBER Y Number of freelist groups allocated in this segment

RELATIVE\_FNO NUMBER Y Relative number of the file containing the segment header

BUFFER\_POOL VARCHAR2(7) Y The default buffer pool to be used for segments blocks

从segment\_type列的comment信息中，可以看出数据段的类型是多样的。任何种类的数据库对象，本质上都是一种数据段。数据表、索引、回滚、聚集这些都是数据段的一种表现形式。同时，数据段是在数据对象创建的时候就已经创建出来，随着对象体积的增大，而不断分配多个extents进行管理。

分区实际上就是存在分开存储的可能。一般一个对象是不会跨物理存储进行存放的，分区表是对应的多个segment。所以，分区表分开存储空间是可能的。

#### 表空间tablespace

TableSpace是存储结构中的最高层结构。建立一个表空间的时候，是需要指定存储的文件。一个表空间可以指定多个数据文件，多个文件可以在不同的物理存储上。也就是说，表空间是可以跨物理存储的。但是有一点就是，表空间下一级对象数据段的存储，是不能指定存储在那个文件里的。所以，要想让数据对象访问IO负载均衡，需要指定不同的数据对象在不同的表空间里。这也就是为什么将数据表和索引建立在不同的表空间的原因。

表空间通过dba\_tablespaces和v$tablespace查询

SQL> desc dba\_tablespaces;

Name Null? Type

----------------------------------------- -------- ----------------------------

TABLESPACE\_NAME NOT NULL VARCHAR2(30)

BLOCK\_SIZE NOT NULL NUMBER

INITIAL\_EXTENT NUMBER

NEXT\_EXTENT NUMBER

MIN\_EXTENTS NOT NULL NUMBER

MAX\_EXTENTS NUMBER

MAX\_SIZE NUMBER

PCT\_INCREASE NUMBER

MIN\_EXTLEN NUMBER

STATUS VARCHAR2(9)

CONTENTS VARCHAR2(9)

LOGGING VARCHAR2(9)

FORCE\_LOGGING VARCHAR2(3)

EXTENT\_MANAGEMENT VARCHAR2(10)

ALLOCATION\_TYPE VARCHAR2(9)

PLUGGED\_IN VARCHAR2(3)

SEGMENT\_SPACE\_MANAGEMENT VARCHAR2(6)

DEF\_TAB\_COMPRESSION VARCHAR2(8)

RETENTION VARCHAR2(11)

BIGFILE VARCHAR2(3)

PREDICATE\_EVALUATION VARCHAR2(7)

ENCRYPTED VARCHAR2(3)

COMPRESS\_FOR VARCHAR2(12)

SQL> desc v$tablespace

Name Null? Type

----------------------------------------- -------- ----------------------------

TS# NUMBER

NAME VARCHAR2(30)

INCLUDED\_IN\_DATABASE\_BACKUP VARCHAR2(3)

BIGFILE VARCHAR2(3)

FLASHBACK\_ON VARCHAR2(3)

ENCRYPT\_IN\_BACKUP VARCHAR2(3)

其中v$tablespace的BIGFILE标志该tablespace是否支持大文件表空间。另一个是flashback\_on，表示表空间的闪回特性是否开启。

#### 表空间tablespace和模式schema之间的关系

数据表的段Segment区Extent块Block的概念，很容易与schema的结构相混淆。schema是一个组织概念，是来自于经典数据库理论范畴。在oracle中，Schema就是一个组织概念，一个user对应的就是一个schema。schema是逻辑对象的集合组织，同表空间等概念不是一个层面的。

在一个schema里，是可以将对象建立在任何数据表空间内的，只有一个默认表空间的概念default tablespace。指定默认表空间是在创建用户的时候指定的。

### 4.3、表空间tablespace和表空间组

#### ASSM

Auto Segment Space Management自动段空间管理

通过ASSM可以在Oracle RAC环境中，不再需要定义空闲列表、空闲列表组等。

查询所有表空间desc dba\_tablespaces;

其中列SEGMENT\_SPACE\_MANAGEMENT，表示该表空间是否在ASSM管理下。

### 4.4、模式（schema）

参考<http://carrie.blog.51cto.com/383855/77617>

参考http://langgufu.iteye.com/blog/1469055

一个用户一般对应一个schema,该用户的schema名等于用户名，并作为该用户缺省schema。这也就是我们在企业管理器的方案下看到schema名都为数据库用户名的原因。用户是不能创建schema的，schema在创建用户的时候创建，并可以指定用户的各种表空间。

### 4.5、模式schema中的逻辑对象

1. **TABLE**   表   用于存储数据的基本结构
2. **VIEW**    视图  以不同的侧面反映表的数据，是一种逻辑上的表
3. **INDEX**   索引  加快表的查询速度
4. CLUSTER 聚簇  将不同表的字段并用的一种特殊结构的表集合
5. **SEQUENCE**    序列  生成数字序列，用于在插入时自动填充表的字段
6. SYNONYM 同义词 为简化和便于记忆，给对象起的别名
7. **DATABASE** LINK   数据库链接   为访问远程对象创建的通道
8. STORED **PROCEDURE**、**FUNCTION**   存储过程和函数 存储于数据库中的可调用的程序和函数
9. PACKAGE、PACKAGE BODY    包和包体    将存储过程、函数及变量按功能和类别进行捆绑
10. **TRIGGER** 触发器 由DML操作或数据库事件触发的事件处理程序

#### 表Table

#### 视图View

#### 索引Index

#### 聚表Cluster

参考http://blog.csdn.net/robinson\_0612/article/details/5925501

簇其实就是一组表，是一组共享相同数据块的多个表组成。 将经常一起使用的表组合在一起成簇可以提高处理效率。

#### 序列Sequence

#### 同义Synonym

#### 数据库链接Database Link

#### 存储过程和函数Stored Procedure、Function

#### 包和包体Package、Package Body

#### 触发器Trigger

# ASM相关

参考http://blog.csdn.net/robinson\_0612/article/details/6080659

## 定义

ASM磁盘

磁盘组

## 操作

### 2.1、ORACLEASM命令

### 2.2、ASMCMD命令