Oracle DB 管理空间

• 介绍Oracle DB Server 如何自动管理空间

• 使用压缩节省空间

• 主动监视和管理表空间的空间使用量

• 介绍Oracle DB 中的段的创建

• 控制延迟创建段

• 使用“段指导”

• 使用段收缩功能从表和索引中回收浪费的空间

• 管理可恢复的空间分配

* **空间管理：概览**

空间由Oracle DB Server 自动管理。Oracle DB Server 可以生成有关潜在问题的预警，并推荐可能的解决方案。

这些功能包括：

• Oracle Managed Files (OMF)

• 用位图进行的空闲空间管理（“本地管理”）和数据文件自动扩展

• 主动的空间管理（默认阈值和服务器生成的预警）

• 空间回收（收缩段、联机重新定义表）

• 容量计划（增长报表）

空间管理：概览

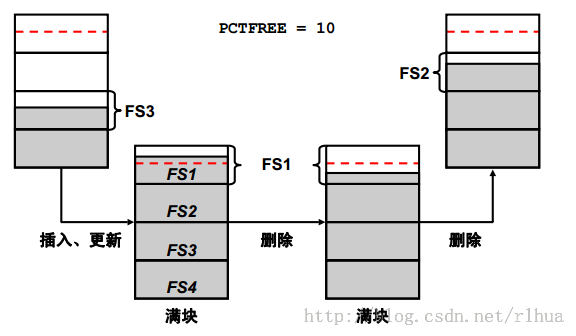
通过使用Oracle Managed Files (OMF)，你可以根据数据库对象（而不是文件名）来指定操作。Oracle DB Server 可以用位图来管理表空间内的空闲空间。这称为“本地管理的”表空间。此外，位于本地管理的表空间中的段内的空闲空间可以使用位图进行管理。这称为“自动段空间管理”。位图化的实施可以避免很多与空间相关的表优化工作，同时能在高峰负载期间提供更高的性能。另外，Oracle DB Server 还可以自动扩展数据文件，因此文件可以根据其中的数据量自动增长。

创建数据库时，默认情况下将启用预先空间监视。（这不会对性能产生任何影响。）

Oracle DB Server 在正常空间分配和取消分配操作期间监视空间使用情况，如果空闲空间的可用性低于预定义的阈值（可以覆盖此阈值），将发出预警。指导和向导可帮助你回收空间。

计划容量时，Oracle DB Server 根据表结构和行数进行空间估计，并根据自动工作量资料档案库(AWR) 中存储的历史空间使用情况来提供增长趋势报表。

* **块空间管理**



块空间管理

空间管理包括块级别的空闲空间管理。使用“自动段空间管理”时，每个块分为四部分，分别称为FS1（空闲空间介于0 到25% 之间）、FS2（25% 到50% 空闲）、FS3（50% 到75% 空闲）和FS4（75% 到100% 空闲）。

块的状态将根据其中空闲空间的级别自动更新。这样，你就可以根据所插入行的长度来判断特定块能否满足插入操作的要求。请注意，“满”状态表示不能再向块中插入。

在示例中，左侧的块是一个FS3 块，因为其空闲空间介于50% 和75% 之间。执行了一些插入和更新语句后，将达到PCTFREE（虚线处），并且不能再向该块中插入新行。

现在该块被视为“满”或FS1 块。块的空闲空间级别一旦降至低于下一部分，就会考虑将该块再次用于插入。在上面这种情况中，空闲空间一旦超过25%，状态即变为FS2。

注：大对象(LOB) 数据类型（BLOB、CLOB、NCLOB和BFILE）不使用PCTFREE存储参数。未压缩块和OLTP 压缩块的默认PCTFREE值为10；基本压缩块的默认PCTFREE值为0。

* **行链接和行迁移**

示例：

• 更新时：行的长度增加，超过了块中的可用空闲空间。

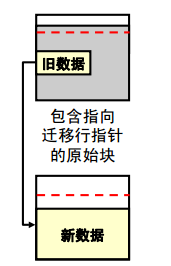
• 需要将数据存储在新块中。

• 将保留行的原始物理标识符(ROWID)。

• Oracle DB Server 需要读取两个块来检索数据。

• “段指导”查找包含迁移行的段。

• 可以对块中的碎片空闲空间进行自动合并。



行链接和行迁移

在以下两种情况下，表中某行的数据可能太大，无法包含在单个数据块中。

第一种情况：

在第一次插入行时，就发现行太大，无法包含在一个数据块中。在这种情况下，Oracle DB Server 将在为该段保留的一系列数据块（一个或多个）中存储该行的数据。较大的行通常要进行行链接，例如包含数据类型为LONG或LONG RAW的列的行。在这些情况下，行链接是无法避免的。

但在第二种情况中，

最初可以包含在一个数据块中的行在更新后，整个行的长度增加了，而块的空闲空间已经完全填满。在这种情况下，Oracle DB Server 假定整个行可以包含在新数据块中，从而将整个行的数据迁移到新数据块中。数据库将保留迁移行的原始行片段，

以便指向包含迁移行的新块。迁移行的ROWID不变。

当行链接或迁移后，与此行关联的输入/输出(I/O) 性能会降低，因为Oracle DB Server 必须扫描多个数据块来检索该行的信息。

“段指导”查找包含由UPDATE操作产生的迁移行的段。

Oracle DB 在以下情况下会以自动透明方式合并数据块的空闲空间：

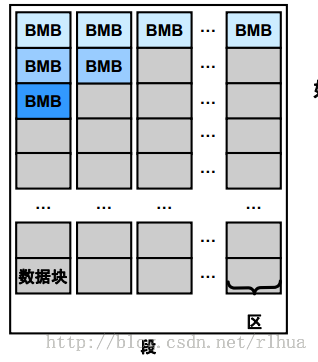
• INSERT或UPDATE语句尝试使用包含足够空闲空间的块来容纳新的行片段

• 空闲空间已成为碎片，以致于行片段无法插入到块的相邻部分

合并之后，空闲空间量与操作前的空间量相同，但空间现在是连续的。

当行链接或迁移后，与此行关联的I/O 性能会降低，因为Oracle DB Server 必须扫描多个数据块来检索该行的信息。

* **段内的空闲空间管理**



段内的空闲空间管理

• 由段中的位图跟踪

好处：

• 空间使用更加灵活

• 运行时调整

•BMB 的多进程搜索

空闲空间管理

可以自动管理数据库段内的空闲空间。段内的空闲空间或已用空间由位图跟踪。要利用此功能，请在创建本地管理的表空间时，指定“自动段空间管理”。然后，指定的设置将应用于随后在此表空间中创建的所有段。

自动空间管理段具有一组位图块(BMB)，用于描述该段中数据块的空间使用情况。BMB 以树形层次结构组织。层次的根层存储在段头部中，其中包含对所有中间层BMB 的引用。

此层次的各个叶节点代表属于该段的一组连续数据块的空间信息。此层次的最大层数为三。

使用自动空间管理的好处包括：

• 改善空间使用情况，尤其是各行大小差异很大的对象的空间使用情况

• 对并行访问发生的变化进行更好的运行时调整

• 在性能方面或空间使用方面改善多实例行为

因此，DBA 可以少做一些工作。

* **段的类型**

段是为某个逻辑结构分配的一组区。段包括以下不同类型：

• 表和簇段

• 索引段

• 还原段

• 临时段

段由Oracle DB Server 动态分配。

段的类型

表和簇段：每个非聚簇表都有一个数据段。所有表数据都存储在表段的区中。对于分区表，每个分区都有一个数据段。每个集群也都有一个数据段。集群中每个表的数据都存储在集群的数据段中。

索引段：每个索引都有一个索引段，存储其所有数据。对于分区索引，每个分区都有一个索引段。

还原段：Oracle DB 会维护用于回退对数据库所做更改的信息。此信息包括事务处理操作的记录；这些操作统称为还原。还原信息存储在还原表空间的还原段中。

临时段：临时段是在SQL 语句需要临时数据库区域来完成执行时由Oracle DB Server 创建的。语句完成执行后，临时段中的区将返回到系统，以备将来使用。

如果段的现有区变满，Oracle DB Server 将动态分配空间。因为区是根据需要分配的，所以段中的区在磁盘上可能连续，也可能不连续。

* **分配区**

• 搜索数据文件的位图，以便获取所需数目的相邻空闲块

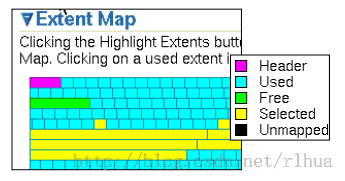
• 可用以下存储子句调整区的大小：

– UNIFORM

– AUTOALLOCATE

• 查看区映射

• 获得取消分配建议



对于本地管理的表空间，Oracle DB Server 将查找要分配给新区的空闲空间，具体方法是：

先在表空间中确定候选数据文件，然后搜索数据文件的位图，以便获取所需数目的相邻空闲块。如果该数据文件没有足够的相邻空闲空间，则Oracle DB Server 将在另一数据文件中查找。

以下两个子句会影响区的大小调整：

• 使用UNIFORM子句，数据库可以按你指定的统一大小（或默认大小）为表空间中创建的所有对象创建所有区。

• 使用AUTOALLOCATE子句，数据库可以为表空间确定区大小调整策略。

要在Oracle Enterprise Manager 中查看区映射，请选择“Server > Tablespaces > View Tablespace > Show Tablespace Contents（服务器> 表空间> 查看表空间> 显示表空间内容）”。

Oracle DB Server 提供了“Segment Advisor（段指导）”，使用该指导可以根据对象中空间碎片的级别来帮助确定对象是否有空间可回收。

* **分配空间**

新的空间分配方法：

• DEFERRED\_SEGMENT\_CREATION = TRUE（默认值）

1. Table creation > Datadictionary operation（创建表> 数据字典操作）

2. DML > Segment creation（DML > 创建段）



Oracle Database 11gR2 包含了一种新的空间分配方法。创建非分区堆表时，表段创建将延迟，直至插入第一行。该功能是默认启用的，即DEFERRED\_SEGMENT\_CREATION初始化参数设置为TRUE。

此新增的空间分配方法的优点包括：

• 对于安装时就会创建成百或上千个表（其中大多数可能永远不会填入内容）的应用程序而言，可以节省大量的磁盘空间。

• 缩短了应用程序安装时间。

在向表中插入第一行后，系统将为基表、其LOB 列及其索引创建段。在创建段期间，表上的游标将失效。这些操作会对性能产生较小的额外影响。

注：使用该新增的分配方法时，务必制定合适的容量计划以便有足够的磁盘空间用于处理填充表时的段创建操作，这一点非常重要。

* **创建没有段的表**

SQL> SHOW PARAMETERS deferred\_segment\_creation

NAME TYPE VALUE

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

deferred\_segment\_creation boolean TRUE

SQL> CREATE TABLE seg\_test(c number, d varchar2(500));

Table created.

SQL> SELECT segment\_name FROM user\_segments;

no rows selected

插入行并创建段：

SQL> INSERT INTO seg\_test VALUES(1, 'aaaaaaa');

1 row created.

SQL> SELECT segment\_name FROM user\_segments;

SEGMENT\_NAME

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

SEG\_TEST

创建没有段的表

本示例显示了如何检查DEFERRED\_SEGMENT\_CREATION参数。然后创建一个没有段的表，可通过查询USER\_SEGMENTS数据字典视图进行验证。插入一行后，再次查询该视图以查看现在是否存在段。

还可以查询USER\_TABLES、USER\_INDEXES或USER\_LOBS视图的SEGMENT\_CREATED列。

对于非分区表、索引和LOB，如果创建了段，该列将显示YES。

数据字典中增加的另一个对象是SYS.SEG$表，该表用于存储在创建表或索引期间指定的存储参数。

* **控制延迟创建段**

在以下项中使用DEFERRED\_SEGMENT\_CREATION参数：

• 初始化文件

• ALTER SESSION命令

• ALTER SYSTEM命令

使用SEGMENT CREATION子句：

• IMMEDIATE

• DEFERRED（Oracle Database 11gR2 中的默认值）

CREATE TABLE SEG\_TAB3(C1 number, C2 number)

SEGMENT CREATION IMMEDIATE TABLESPACE SEG\_TBS;

CREATE TABLE SEG\_TAB4(C1 number, C2 number)

SEGMENT CREATION DEFERRED;

注：索引继承表的特征。

控制延迟创建段

可通过以下两种方式控制段的创建：

• 将DEFERRED\_SEGMENT\_CREATION初始化参数设置为TRUE或FALSE。可在初始化文件中设置该参数。还可以通过ALTER SESSION或ALTER SYSTEM命令进行控制。

示例：

ALTER SESSION SET DEFERRED\_SEGMENT\_CREATION = TRUE;

ALTER SYSTEM SET DEFERRED\_SEGMENT\_CREATION = FALSE;

• 使用CREATE TABLE命令的SEGMENT CREATION子句：

- SEGMENT CREATION DEFERRED：如果指定，则系统将延迟创建段直至表中插入第一行。这是Oracle Database 11gR2 的默认行为。

- SEGMENT CREATION IMMEDIATE：如果指定，则在表创建期间对段进行实体化。这是早于Oracle Database 11gR2 的Oracle DB 中的默认行为。

该子句优先于DEFERRED\_SEGMENT\_CREATION参数。

可以使用ALTER TABLE … MOVE命令为已创建的表强制创建段。

不过，对于从属对象（如索引），无法直接控制延迟创建段。它们继承其父对象（在本例中为表）的这一特征。

* **限制和例外**

按需创建段：

• 仅适用于非分区表和索引

• 不适用于IOT、聚簇表或其它特殊表

• 不适用于字典管理的表空间中的表

注：如果计划将没有段的表从本地管理的表空间迁移到字典管理的表空间，则必须将其删除并重新进行创建。

限制和例外

• 在Oracle Database 11gR2 中，延迟创建段被限制于非分区表和非分区索引。不支持IOT 和其它特殊表。

• 对于聚簇表和在字典管理的表空间中创建的表，不支持按需创建段。如果尝试这样做，系统将创建段。

• 如果在本地管理的表空间中创建了一个采用延迟创建段的表，则该表没有段。如果稍后将该表空间迁移到字典管理的表空间，则任何创建段的尝试都将产生错误。此时，必须将该表删除并重新进行创建。

• 对于聚簇表、全局临时表、特定于会话的临时表、内部表、类型化表、AQ 表、SYS拥有的表、外部表、位图联接索引以及域索引，不支持按需创建段。归SYSTEM、PUBLIC、OUTLN以及XDB所有的表也不在支持范围内。

* **其它自动功能**

没有用户干预：

• 不能用的索引和索引分区没有段

• 创建没有段的索引：

CREATE INDEX test\_i1 ON seg\_test(c) UNUSABLE;

• 删除索引的任何已分配空间：

ALTER INDEX test\_i UNUSABLE;

• 为索引创建段：

ALTER INDEX test\_i REBUILD;

*SELECT segment\_name, partition\_name,*

*segment\_type*

*FROM user\_segments*

*WHERE segment\_name like '%DEMO';*

其它自动功能

Oracle Database 11g发行版2 中还实施了其它增强功能（与延迟创建段无关）以节省空间：创建的所有UNUSABLE索引和索引分区都没有段。该功能对你是完全透明的。该功能默认为启用，即COMPATIBILITY初始刷参数设置为11.2.0.0。

示例：如果你有一个DEMO表，它有三个分区和一个本地索引，则在执行查询时会看到三个表段和三个索引段。

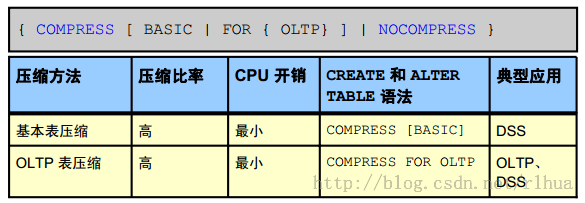
如果将一个表分区移至新的表空间，然后执行同样的查询，你将看到三个表段和两个索引段，这是因为不可用的一个已经自动删除。

* **表压缩：概览**

通过压缩所有数据减少存储成本：

• 基本压缩适用于直接路径插入操作：10x

• OLTP 压缩适用于所有DML 操作：2–4x



典型应用 压缩比率

{ COMPRESS[ BASIC | FOR { OLTP} ] | NOCOMPRESS }

表压缩：概览

Oracle DB 支持以下三种表压缩方法：

• 基本表压缩

•OLTP 表压缩

• Hybrid Columnar 压缩（通过Exadata）

Oracle Corporation 建议压缩所有数据以减少存储成本。Oracle DB 可使用表压缩来消除数据块中的重复值。对于具有高度冗余数据的表，压缩可节省磁盘空间并减少数据库缓冲区高速缓存中的内存使用。表压缩对数据库应用程序是透明的。

• table\_compression 子句仅对按堆组织的表有效。COMPRESS关键字用于启用表压缩。

NOCOMPRESS关键字用于禁用表压缩。NOCOMPRESS为默认值。

• 通过基本压缩，Oracle DB 在使用诸如直接加载或CREATE TABLE AS SELECT等操作执行批量加载时压缩数据。

• 通过COMPRESS FOR OLTP，Oracle DB 在对表执行所有DML 操作期间压缩数据。

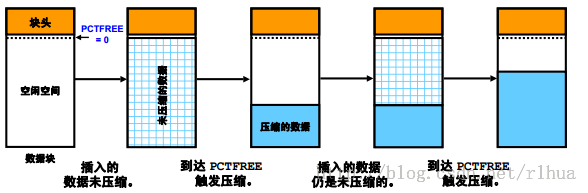
* **适用于直接路径插入操作的压缩**

• 是使用CREATE TABLE …COMPRESS BASIC…;启用的

• 建议为批量加载数据仓库使用该压缩方法

• 替换已废弃的COMPRESS FOR DIRECT\_LOAD OPERATIONS

• 最大化块中的连续空闲空间



适用于直接路径插入操作的压缩

使用COMPRESS或COMPRESS BASIC可启用基本的表压缩。

• Oracle DB 在执行下列直接路径插入操作期间尝试压缩数据（如果这样做有利）：

- 直接路径SQL\*Loader

- CREATE TABLE AS SELECT语句

- 并行INSERT语句

- 带有APPEND提示的INSERT语句

• 原始导入实用程序(imp) 不支持直接路径INSERT，因此无法以压缩的格式导入数据。

• 在早期版本中，这种类型的压缩称为DSS 表压缩，是使用COMPRESS FOR DIRECT\_LOAD OPERATIONS启用的。该语法已废弃。

• 压缩消除了块中因删除操作而形成的空隙，使块中连续可用空间达到最大化。

该幻灯片显示了压缩表中的数据块的演变过程，应按从左到右的顺序阅读。开始的时候，该数据块是空的，可以插入数据。开始在该块中插入数据时，数据以未压缩的格式存储（就像在未压缩的表中一样）。不过，只要该块达到其PCTFREE设置规定的填满标准，数据就会自动进行压缩，以减少其原来占据的空间。

这样一来，可以在同一块中插入新的未压缩数据，直到再一次达到PCTFREE设置规定的填满标准。此时，会再一次触发压缩以减少块中的已使用空间量。

注：采用COMPRESS或COMPRESS BASIC的表使用PCTFREE值0以最大化压缩，除非显式设置PCTFREE子句的值。

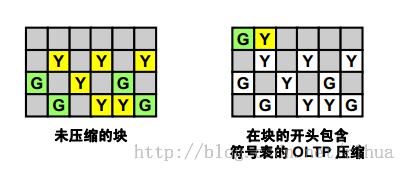
采用COMPRESS FOR OLTP或NOCOMPRESS的表使用PCTFREE的默认值10以最大化压缩，同时仍允许以后对数据进行一些DML 更改，除非显式覆盖该默认值。

* **适用于DML 操作的OLTP 压缩**

• 是使用CREATE TABLE … COMPRESS FOR OLTP…;启用的

• 建议为活动的OLTP 环境使用该压缩方法

• 替换已废弃的COMPRESS FOR ALL OPERATIONS



适用于DML 操作的OLTP 压缩

使用COMPRESS FOR OLTP可启用OLTP 表压缩。

• Oracle DB 在对表执行所有DML 操作期间压缩数据。建议为活动的OLTP 环境使用这种形式的压缩。

• 在早期版本中，OLTP 表压缩是使用COMPRESS FOR ALL OPERATIONS启用的。该语法已废弃。

使用OLTP 压缩，一个数据块中行和列中的重复值只在块的开头存储一次（存储在符号表中）。重复值被替换为对符号表的简短引用（如图中所示）。因此，重新创建未压缩的数据所需的信息存储在块中。

为了说明OLTP 压缩的原理，示例中的图表显示了两个矩形。第一个灰色矩形包含四个标记为“G”的绿色小正方形和六个标记为“Y”的黄色正方形。它们表示未压缩的块。

在第二个灰色矩形的开头，仅有一个标记为“G”的绿色正方形和一个标记为“Y”的黄色正方形，表示符号表。第二个灰色图表在与绿色和黄色正方形相同的位置还显示了10 个白色的正方形。它们是白色的，因为它们现在仅是引用，重复值不占用空间。

* **指定表压缩**

可以为以下项指定表压缩：

• 整个按堆组织的表

• 分区表（每个分区可具有一种不同类型或级别的压缩。）

• 嵌套表的存储

你无法：

• 为列数多于255 的表指定基本压缩和OLTP 压缩

• 如果表是针对直接加载而压缩的，则你无法删除列；但如果表使用OLTP 压缩，则可删除

指定表压缩

可以为以下项指定表压缩：

• 整个按堆组织的表（在relational\_table 或object\_table 的physical\_properties 子句中）

• 分区表（每个分区可具有一种不同类型或级别的压缩。）

• 嵌套表的存储（在nested\_table\_col\_properties 子句中）

表压缩具有以下限制：

• 不支持对列数多于255 的表使用COMPRESS FOR OLTP和COMPRESS BASIC

• 无法从针对直接加载操作压缩的表中删除列，不过可以将这类列设置为未使用。

ALTER TABLE... drop\_column\_clause 的所有操作对使用OLTP 压缩的表有效

• 使用Hybrid Columnar 压缩的表不支持逻辑备用、Streams 和LogMiner

* **使用压缩指导**

压缩指导：

• 分析对象，估计不同压缩方法所节省的空间

• 帮助确定应用程序的正确压缩级别

• 推荐各种压缩策略

– 为特定的数据集选取正确的压缩算法

– 按特定列进行排序，以提高压缩比率

– 提供各种压缩算法的折衷方案

• 适用于OLTP 压缩（通过EM）

使用压缩指导

压缩指导分析数据库对象，并确定每个压缩级别可实现的预期压缩比率。因此，它可帮助你确定应用程序的正确压缩级别。该指导推荐各种压缩策略。当从EM 访问压缩指导时，它将确定OLTP 压缩。

* **使用DBMS\_COMPRESSION程序包**

确定最佳压缩比率：

BEGIN

DBMS\_COMPRESSION.**GET\_COMPRESSION\_RATIO** ('USERS','SH','SALES',

NULL,DBMS\_COMPRESSION.COMP\_FOR\_OLTP, blkcnt\_cmp, blkcnt\_uncmp,

rowcnt\_cmp, rowcnt\_uncmp, comptype);

DBMS\_OUTPUT.PUT\_LINE('Blk count compressed = ' || blkcnt\_cmp);

DBMS\_OUTPUT.PUT\_LINE('Blk count uncompressed = ' ||

blkcnt\_uncmp);

DBMS\_OUTPUT.PUT\_LINE('Row count per block compressed = ' ||

rowcnt\_cmp);

DBMS\_OUTPUT.PUT\_LINE('Row count per block uncompressed = ' ||

rowcnt\_uncmp);

DBMS\_OUTPUT.PUT\_LINE('Compression type = ' || comptype);

DBMS\_OUTPUT.PUT\_LINE('Compression ratio =

'||blkcnt\_uncmp/blkcnt\_cmp||' to 1');

使用DBMS\_COMPRESSION程序包

DBMS\_COMPRESSION程序包提供的压缩指导帮助你为指定表确定可预期的压缩比率。

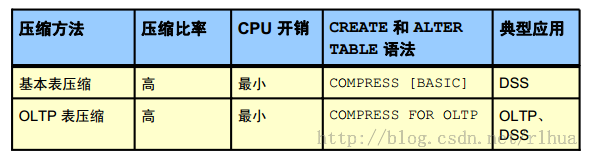
该指导将分析数据库中的对象，确定可能达到的压缩比率，并建议最佳压缩级别。除了DBMS\_COMPRESSION程序包之外，还可以在现有指导框架内使用该压缩指导（通过DBMS\_ADVISOR程序包）。

为了确定压缩比率，DBMS\_COMPRESSION程序包提供了以下子程序：

• GET\_COMPRESSION\_RATIO过程，可针对未压缩表给出可行的压缩比率。

• GET\_COMPRESSION\_TYPE过程，针对给定行返回压缩类型。

* **压缩表数据**



Oracle DB 支持以下三种表压缩方法：

• 基本表压缩

•OLTP 表压缩

• Hybrid Columnar 压缩（通过Exadata）

* **预先表空间监视**

数据库以下列方式预先管理表空间的磁盘空间使用量：

• 表空间的可用磁盘空间变低，以及特定段空间不足时，系统会通过数据库预警通知你。

然后你就可以为表空间提供更多磁盘空间，以避免空间不足的情况发生。

• 收集的信息被存储在自动工作量资料档案库(AWR) 中，用于执行增长趋势分析以及计划数据库的容量。

要在Oracle Enterprise Manager 中查看和修改表空间信息，请在“Database（数据库）”主页中选择“Server（服务器）”，然后选择“Tablespaces（表空间）”。选择需要的表空间，然后单击“Edit（编辑）”按钮。

* **阈值和解决空间问题**



通过以下方法解决空间问题：

• 添加数据文件或调整数据文件大小

• 将AUTOEXTEND设置为ON

• 收缩对象

• 减少UNDO\_RETENTION

• 检查临时表空间中是否存在长时间运行的查询

阈值和解决空间问题

表空间阈值定义为“满”，或者定义为表空间中的可用空间。严重阈值和警告阈值是应用于表空间的两个阈值。DBMS\_SERVER\_ALERT程序包中包含用于设置和获取阈值的过程。

达到表空间限制大小时，将引发相应的预警。阈值以表空间大小的百分比，或剩余的空闲字节数表示。此值在内存中计算。可以为表空间同时定义百分比和字节形式的阈值。它们中的任何一个或两者都可以生成预警。

理想的警告阈值触发值设置会生成这样的预警：既能确保有足够时间来解决问题，使其不会变成严重状态，又不至于在空间尚未成为问题时干扰你。

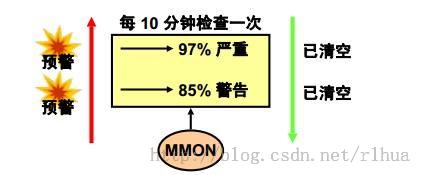
预警指示DBA 可以通过执行以下一项或多项操作来解决问题：

• 向表空间添加更多空间，方法是添加一个文件或调整现有文件的大小，或者使现有文件可自动扩展

• 在包含任何可自动扩展文件的磁盘上释放空间

• 收缩表空间中的稀疏对象

* **监视表空间的空间使用量**



• 只读表空间和脱机表空间：不要设置预警。

• 临时表空间：阈值对应于会话当前使用的空间。

• 还原表空间：阈值对应于活动区和未到期区所使用的空间。

• 可自动扩展的文件：阈值基于最大文件大小。

监视表空间的空间使用量

数据库在执行常规的空间管理活动时跟踪空间使用情况。MMON进程每10 分钟汇总一次此信息。达到或超过了表空间的阈值时，将触发预警。

• 不应在处于只读模式的表空间或已脱机的表空间上标记预警，因为在这些表空间上无法进行太多操作。

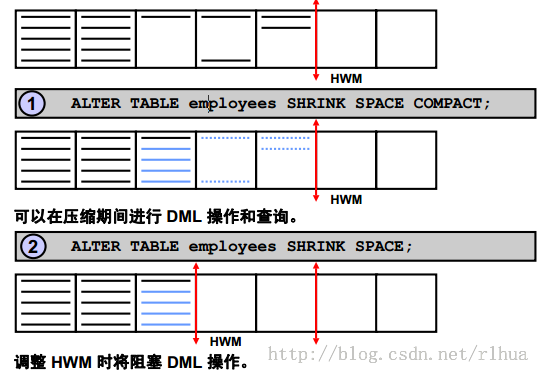
• 在临时表空间中，阈值必须定义为对表空间中已用空间的限制。

• 对于还原的表空间，如果某个区中不包含活动或未到期的还原操作，则该区可以重用。

计算阈值违规时，活动区和未到期区总体被视为已用空间。

• 对于包含可自动扩展文件的表空间，将根据你指定的最大文件大小或最大OS 文件大小来计算阈值。

* **收缩段**



图表描述了表收缩操作的两个阶段。第一个阶段执行压缩。在此阶段，行将尽可能移动到段的左侧部分。在内部，行将由数据包移动，以避免锁定问题。移动完行后，将启动收缩操作的第二阶段。在此阶段中，将调整高水位标记(HWM)，并释放未使用的

空间。

如果有长时间运行的查询，而且这些查询可能跨越收缩操作，并尝试从已回收的块中读取数据，则COMPACT子句会很有用。指定SHRINK SPACE COMPACT子句时，收缩操作的进度将保存在相应段的位图块中。这意味着下次在同一个段上执行收缩操作时，Oracle DB Server 可以记住已经执行过的操作。然后你就可以在非峰值时段重新发出SHRINK SPACE子句，无需使用COMPACT子句即可完成第二阶段。

* **收缩操作的结果**

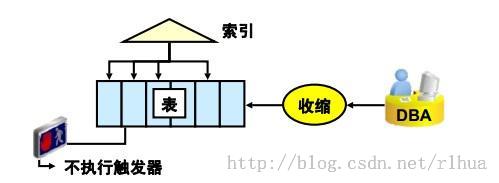
• 改善性能和空间使用情况

• 维护索引

• 不执行触发器

• 可以减少迁移行的数量

• 建议在IOT 上重建二级索引



收缩稀疏填充的段可以提高对该段执行扫描操作和DML 操作的性能。这是因为在收缩段后，需要查看的块减少了。尤其体现在下列方面：

• 全表扫描（块变少、变密）

• 改善索引访问（由于树变得更紧凑，因此减少了ROWID范围扫描时的I/O 次数）

另外，通过收缩稀疏填充的段，还可以提高数据库内空间使用效率，因为在对象需要空间时有更多空闲空间可以使用。

在段收缩操作期间将保证索引依赖关系。收缩了相应表后，索引处于可用状态。因此，不需要进一步维护。

实际收缩操作在内部作为INSERT/DELETE操作进行处理。但是，不执行任何DML 触发器，因为数据本身并未发生变化。

段收缩操作的一个可能结果是迁移行的数量减少。但是，你不应总是依赖段收缩来减少迁移行数量。因为段收缩操作可能不会触及段中的所有块。因此，不能保证所有迁移行都得到处理。

注：建议执行收缩操作之后在索引表(IOT) 上重建二级索引。

* **回收ASSM 段内的空间**

• 联机原地操作

• 只适用于位于ASSM 表空间中的段

• 候选段类型包括：

– 按堆组织的表和索引表

– 索引

– 分区和子分区

– 实体化视图和实体化视图日志

使用ASSM 回收空间

收缩操作是一个联机原地操作，因为不需要额外的数据库空间即可执行此操作。

• 无法在由空闲列表管理的段上执行收缩操作。可以收缩自动段空间管理的表空间中的段。但是，不能收缩ASSM 表空间中存储的下列对象：

- 集群中的表

- 包含LONG列的表

- 包含提交时实体化视图的表

- 包含基于ROWID的实体化视图的表

-IOT 映射表

- 包含基于函数的索引的表

• 必须为按堆组织的段启用ROW MOVEMENT。

* **段指导：概览**

段指导可标识具有可以回收的空间的段。它通过检查自动工作量资料档案库(AWR) 中的使用情况和增长统计数据以及对段中的数据进行采样来执行分析。可以将其配置为定期自动运行，也可以根据需要（手动）运行它。定期调度的段指导运行称为“自动段指导”。提供建议之后，你可以选择实施这些建议。可以在段级别或表空间级别调用收缩指导。

EM 数据库控制台是段指导的接口。可以从EM 中的几个位置访问段指导：

• “Advisor Central（指导中心）”页

• “Tablespaces（表空间）”页

• 方案对象页

使用数据库控制台可以选择不同的输入信息，并可以调度一个调用段指导的作业来获得收缩建议。可以在表空间环境中或方案对象环境中调用段指导向导，没有相关环境也可以调用。

段指导可以根据采样分析、历史信息和未来增长趋势来提供建议。

* **段指导**

在“Server（服务器）”页中的“Storage（存储）”部分，选择“Tablespaces（表空间）”。

在“Tablespaces（表空间）”页上，选择要对其执行收缩分析的表空间，然后在“Actions（操作）”下拉列表中选择“Run Segment Advisor（运行段指导）”。单击“Go（执行）”

可以打开“Segment Advisor（段指导）”初始页。必须从“comprehensive（综合）”和“limited（有限制）”分析模式中选择一种。在综合模式下，分析时间较长，因为指导会对段进行采样以确定正确的目标。

一直单击“Continue（继续）”，回答指导的各个问题。你在“Segment Advisor: Review（段指导: 复查）”页上结束操作，可以在该页上复查分析内容的详细资料。段指导分析作为调度作业运行，因此可以从“Advisor Central（指导中心）”页检查已调度的任务。完成后，可以检查指导的建议。

注：在段指导中，可以指定分析操作的持续时间。这样，可以限制指导用于生成建议的时间。一般说来，分析时段越长，得到的结果就越全面。结果存储在AWR 中，可以稍后查看。使用“Number of days to retain（保留天数）”选项可以告知Oracle DB Server 这些结果应在AWR 中保留多少天后再清除。

* **实施建议**

段指导完成其作业后，你可以查看建议的详细资料并直接实施建议。

注：在收缩按堆组织的表之前，必须在该表上启用行移动功能。可以使用“Edit Table（编辑表）”页上“Options（选项）”选项卡中的“Database Control（数据库控制）”来进行此操作。

* **自动段指导**

• 由设置为在默认维护窗口期间运行的调度程序作业启动：

– 每个工作日的晚上，星期一至星期五，从晚上10:00 至次日凌晨2:00

– 星期六和星期日，两个窗口都开始于早上6:00，并持续20 小时

• 检查数据库统计信息，对段数据采样，然后选择以下对象进行分析：

– 超过了严重或警告阈值的表空间

– 包含活动最多的段

– 增长率最高的段

自动段指导

自动段指导是由配置为在默认维护窗口期间运行的调度程序作业启动的。默认维护窗口是在调度程序中指定的，其初始定义如下：

• 每个工作日的晚上，星期一至星期五，从晚上10:00 至次日凌晨2:00（每晚4 小时）

• 周末，星期六和星期日早上6:00，每天持续20 小时

自动段指导不会分析每个数据库对象。而是检查数据库统计信息，对段数据采样，然后选择以下对象进行分析：

• 超过了严重或警告空间阈值的表空间

• 包含活动最多的段

• 增长率最高的段

如果选择了某个对象进行分析，但维护窗口在段指导能处理该对象之前失效了，则该对象将在下一次自动段指导运行中分析。不能更改自动段指导选择进行分析的那组表空间和段。

但是，可以启用或禁用自动段指导作业，更改自动段指导的计划运行期间，或者调整自动段指导的系统资源利用率。

* **使用EM 手动收缩段**

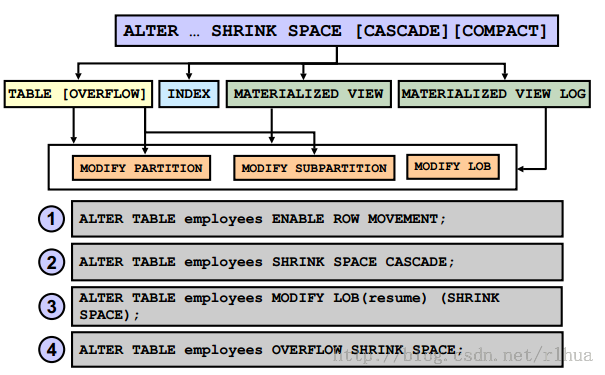
使用EM 手动收缩段

此外（相对于实施段指导的建议），还可以收缩与特定数据库对象关联的各个段。例如，在“Database（数据库）”主页中选择“Schema（方案）”文件夹选项卡，然后单击“Database Objects（数据库对象）”部分中的“Tables（表）”链接。在“Tables（表）”页上选择表，接着在“Actions（操作）”下拉列表中选择“Shrink Segment（收缩段）”。

然后单击“Go（执行）”按钮。这样将打开“Shrink Segment（收缩段）”页，你可以在其中选择要收缩的从属段。可以选择只压缩空间，或者选择压缩并释放空间。还可以选择CASCADE选项。

完成后，单击“Continue（继续）”链接。这样，收缩语句将作为调度作业提交。

* **使用SQL 收缩段**



使用SQL 收缩段

因为在按堆组织的段中，收缩操作可能会导致ROWID发生更改，所以在该段上执行收缩操作之前，必须在相应段上启用行移动。默认情况下，在段级别上行移动处于禁用状态。

要启用行移动，需要使用CREATE TABLE或ALTER TABLE命令的ENABLE ROW MOVEMENT子句。第一个示例对此进行了阐述。

使用ALTER命令可以针对对象调用段收缩。对象可以是下列类型之一：表（按堆组织的表或索引表）、分区、子分区、LOB（数据和索引段）、索引、实体化视图或实体化视图日志。

使用SHRINK SPACE子句可以收缩段中的空间。如果指定了CASCADE，则收缩行为将级联到所有支持收缩操作的从属段，但实体化视图、LOB 索引和IOT（索引表）映射表例外。第二个示例展示了SHRINK SPACE子句。

在索引段中，收缩操作将首先合并索引，然后压缩数据。示例3 显示了一个收缩LOB 段的命令，并假设RESUME列为CLOB。

示例4 显示了一个收缩IOT 溢出段的命令，此段属于EMPLOYEES表。

。

* **管理可恢复的空间分配**

可恢复的语句有下列特性：

• 使用可恢复的语句，可以挂起大型操作，而不是收到错误

• 使用可恢复的语句，当操作挂起时，你可以解决问题，然后继续进行，无须重新开始

• 在下列条件下，可恢复的语句将挂起：

– 空间不足

– 达到了最大区数

– 超出了空间限额

• 可恢复的语句可以多次挂起和恢复

管理可恢复的空间分配

Oracle DB Server 提供了一种方法，可以在空间分配失败时挂起大型数据库操作，稍后再恢复执行。使用这种方法，就有机会采取更正措施，而不是让Oracle DB Server 向用户返回错误。更正了错误条件后，挂起的操作将自动恢复。这种功能称为“可恢复的空间分配”。受影响的语句称为“可恢复的语句”。仅当为系统或会话启用了可恢复的语句功能时，语句才能在可恢复模式下执行。

挂起语句时将自动挂起事务处理。因此在SQL 语句的挂起和恢复过程中，将保留所有事务处理资源。错误条件不再存在时（例如在用户干预之后，或者很可能在其它查询释放了排序空间之后），挂起的语句将自动恢复执行。出现下列条件之一时，可恢复的语句将挂起：

• 空间不足条件

• 达到了最大区数条件

• 超出了空间限额条件

可恢复的语句有一个与之关联的挂起超时间隔。可恢复的语句的挂起时间超过超时间隔（默认为2 小时）后将重新激活自身，并向用户返回异常错误。可恢复的语句可以多次挂起和恢复。

注：达到最大区数错误仅发生在字典管理的表空间中。

* **使用可恢复的空间分配**

• 如果查询、DML 操作和特定DDL 操作遇到空间不足错误，则可恢复这些操作。

• 可以通过SQL、PL/SQL、SQL\*Loader 和数据泵实用程序或Oracle 调用接口(OCI) 来发出可恢复的语句。

• 仅当某条语句的会话已由下列操作之一启用时，该语句才可以在可恢复模式下执行：

– RESUMABLE\_TIMEOUT初始化参数被设置为非零值。

– 发出了ALTER SESSION ENABLE RESUMABLE语句：

*ALTER SESSION ENABLE RESUMABLE;*

*INSERT INTO sales\_new SELECT \* FROM sh.sales;*

*ALTER SESSION DISABLE RESUMABLE;*

使用可恢复的空间分配

仅当在启用了可恢复模式的会话中执行语句时，才能使用可恢复的空间分配。有两种方法可以启用和禁用可恢复的空间分配：

• 发出ALTER SESSION ENABLE RESUMABLE命令。

• 使用ALTER SESSION或ALTER SYSTEM语句，将RESUMABLE\_TIMEOUT初始化参数设置为非零值。

在为会话或数据库启用可恢复模式时，可以指定一个超时时段，在此时段后，如果未发生任何干预，则挂起的语句将出错。RESUMABLE\_TIMEOUT初始化参数指示超时发生之前经过的秒数。也可以用以下命令来指定超时时段：

*ALTER SESSION ENABLE RESUMABLE TIMEOUT 3600;*

TIMEOUT值一直有效，直到另一个ALTER SESSION ENABLE RESUMABLE语句更改该值，或使用另一种方式更改该值，或直到会话结束。使用ENABLE RESUMABLETIMEOUT子句启用可恢复模式时的默认超时间隔是7,200 秒（2 小时）。

还可以为可恢复的语句命名。例如：

*ALTER SESSION ENABLE RESUMABLE TIMEOUT 3600*

*NAME 'multitab insert';*

语句的名称用于在DBA\_RESUMABLE和USER\_RESUMABLE视图中标识可恢复的语句。

例如：

*SELECT name, sql\_text FROM user\_resumable;*

*NAME SQL\_TEXT*

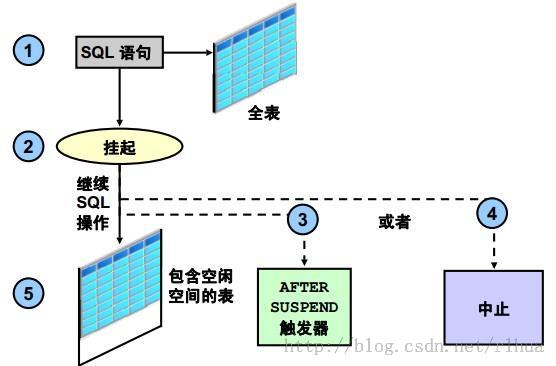
*--------------- --------------------------------------------*

*multitab insert INSERT INTO oldsales SELECT \* FROM sh.sales;*

要自动为各个会话配置可恢复的语句设置，可以创建并注册一个数据库级别的LOGON触发器，以更改用户的会话。此触发器可以发出命令来为会话启用可恢复的语句，指定超时时段，以及将会话发出的可恢复语句与某个名称关联。

因为挂起的语句会保留一些系统资源，所以必须先向用户授予RESUMABLE系统权限，然后他们才能启用可恢复的空间分配并执行可恢复的语句。

* **恢复挂起的语句**



示例

1.INSERT语句遇到错误，声称表已满。

2.挂起INSERT语句，并且不向客户机传递任何错误。

3.或者，执行一个AFTER SUSPEND触发器。

4.或者，激活SQLERROR异常错误以中止语句。

5.如果语句未中止，并将空闲空间成功添加到表中，则INSERT语句将恢复执行。

检测挂起的语句

可恢复的语句挂起时，不会向客户机发出错误。为了方便采取更正措施，Oracle DB Server 提供了无需向用户通知错误，也无需提供有关情况的信息的替代方法。

**在挂起期间可能执行的操作**

可恢复的语句遇到可更正的错误时，系统将在内部生成AFTER SUSPEND系统事件。用户可以在数据库级别和方案级别同时为此事件注册触发器。如果用户注册一个触发器来处理此系统事件，则在SQL 语句挂起后将执行该触发器。在AFTER SUSPEND触发器内执行的SQL 语句始终不可恢复，并且始终是独立运行的。在触发器内启动的事务处理使用SYSTEM回退段。限定这些条件是为了克服死锁，并降低触发器与语句遇到相同错误条件的概率。

在触发器代码中，可以使用USER\_RESUMABLE视图或DBA\_RESUMABLE视图，或者使用DBMS\_RESUMABLE.SPACE\_ERROR\_INFO函数来获得有关可恢复语句的信息。

当可恢复的语句挂起时：

• 调用语句的会话将被置于等待状态。对于EVENT列包含“statement suspended, wait error to be cleared”的会话，系统会将一个行插入V$SESSION\_WAIT。

• 针对需要添加资源才能完成挂起语句的对象发出操作挂起的预警。

**结束挂起的语句**

解决了错误条件后（例如在DBA 干预之后，或者很可能在其它查询释放了排序空间之后），挂起的语句将自动恢复执行，并且清除“可恢复会话已挂起”预警。

可以使用DBMS\_RESUMABLE.ABORT()过程强制挂起的语句激活SERVERERROR异常错误。此过程可以由DBA 调用，也可以由发出语句的用户调用。如果达到了与可恢复语句关联的挂起超时间隔，则语句将自动中止并且系统会向用户返回错误。

* **哪些操作是可恢复的**

下列操作是可恢复的操作：

• 查询：运行时临时空间不足（排序空间不足）的SELECT语句是可恢复执行的语句。

使用OCI 时，OCIStmtExecute()和OCIStmtFetch()调用也可恢复执行。

• DML：INSERT、UPDATE和DELETE语句是可恢复执行的语句。用于执行这些语句的接口没有影响；接口可以是OCI、SQL、PL/SQL 或其它接口。此外，外部表中的INSERT INTO ... SELECT也是可恢复的。

• DDL：下列语句是可恢复执行的操作：

- CREATE TABLE ... AS SELECT

- CREATE INDEX

- ALTER INDEX ... REBUILD

- ALTER TABLE ... MOVE PARTITION

- ALTER TABLE ... SPLIT PARTITION

- ALTER INDEX ... REBUILD PARTITION

- ALTER INDEX ... SPLIT PARTITION

- CREATE MATERIALIZED VIEW

* **小结**

• 介绍Oracle DB Server 如何自动管理空间

• 使用压缩节省空间

• 主动监视和管理表空间的空间使用量

• 介绍Oracle DB 中的段的创建

• 控制延迟创建段

• 使用“段指导”

• 使用段收缩功能从表和索引中回收浪费的空间

• 管理可恢复的空间分配