

表領域の領域管理

目次

はじめに

1. 現状の領域使用率を確認するための SELECT 文
 2. 主な拡張エラー番号についての紹介
 3. 拡張エラーに対する対処
 4. 領域管理に関する FAQ
 5. ローカル管理の表領域とディクショナリ管理の表領域について
- 最後に

はじめに

既表や索引などのセグメントは多くの領域を必要とするため、領域の監視と拡張エラーを防ぐための対処が重要です。今回は表領域内の使用状況を確認するための SQL や、領域管理について頂く質問内容をもとにした FAQ などを紹介させていただきます。

注) 動作確認環境

OS : Windows 2000 Professional SP4
Oracle : Oracle9i R9.0.1.4

1. 現状の領域使用率を確認するための SELECT 文

まず始めに、現在の領域の使用状況を確認するため各表領域の空き状況を求める SQL をご紹介します。

```
select  d.tablespace_name,"file_size(M)" , "max_freesize(M)",
        max_freeblocks , "sum_freesize(M)"
from    (select tablespace_name,sum(bytes)/(1024*1024) as "file_size(M)"
        from dba_data_files
        group by tablespace_name) d,
        (select tablespace_name,max(bytes)/(1024*1024) as "max_freesize(M)",
        sum(bytes)/(1024*1024) as "sum_freesize(M)",max(blocks) as max_freeblocks
        from dba_free_space
        group by tablespace_name) f
where f.tablespace_name=d.tablespace_name
order by 2 desc ,3 desc;
```

上記の SQL は SYS、SYSTEM などの管理ユーザとしてログインして実行してください。
検索結果の例をご紹介します。

TABLESPACE	file_size(M)	max_freesize(M)	MAX_FREEBLOCKS	sum_freesize(M)
UNDOTBS	1450	1227.625	314272	1445.6875
USERS	660	63	16128	446.3125
SYSTEM	325	133.933594	34287	134.226563
TOOLS	30	.9375	240	.9375
INDX	25	24.9375	6384	24.9375
CWMLITE	20	19.9375	5104	19.9375
TEST	1	.91796875	235	.91796875

主な列の意味は以下のとおりです。

- ・ file_size(M) : 表領域の合計サイズ (MB 単位)
- ・ max_freesize(M) : 各表領域の連続している最大の空き領域 (MB 単位)
- ・ max_freeblocks : 各表領域の連続している最大の空きブロック数
- ・ sum_freesize(M) : 各表領域の合計空き領域 (MB 単位)

エクステントを表領域内に追加するためには、そのエクステントサイズ分の連続した空き領域が必要です。連続した空き領域が獲得できなかった場合、領域不足のエラーが発生します。

次に、表領域内の各セグメントのサイズおよびセグメントが占める割合を求める SQL をご紹介します。

```
select s.tablespace_name,s.owner,s.segment_name,s.segment_type,
       s.mbytes,round(s.mbytes/d.mbytes*100,2) as "used%"
from
  (select tablespace_name,owner,segment_name,segment_type,bytes/(1024*1024) as Mbytes
   from dba_segments) s,
  (select tablespace_name,sum(bytes)/(1024*1024) as mbytes
   from dba_data_files
   group by tablespace_name) d
where s.tablespace_name=d.tablespace_name
and d.tablespace_name IN ('&表領域名')
order by s.mbytes desc,s.owner;
```

SYS、SYSTEM などの管理ユーザとしてログインして実行してください。

また、SQL*Plus の置換変数「&」を使用して表領域名を指定させています。全ての表領域内のセグメント情報を参照したい場合は、「and d.tablespace_name IN ('&表領域名')」を削除してください。

検索結果の例を記載します。

表領域名に値を入力してください: USERS

旧 10: and d.tablespace_name IN ('&表領域名')

新 10: and d.tablespace_name IN ('USERS')

TABSPACE	OWNER	SEGMENT_NAME	SEGMENT_TY	MBYTES	used%
USERS	MASTER	BONUS	TABLE	150	22.73
USERS	MASTER	SYU_EMPNO_B	INDEX	62	9.39
USERS	SCOTT	SALGRADE	TABLE	.375	.06
USERS	ALLEN	SYU_EMPNO	INDEX	.0625	.01
USERS	ALLEN	SYU_DEPTNO	INDEX	.0625	.01

主な列の意味は以下のとおりです。

- ・ MBYTES : セグメントのサイズ (MB 単位)
- ・ used% : セグメントが表領域内で占める割合

例えば、サイズの大きいセグメントが拡張する時には多くの空き領域が必要となります。
この時表領域内に十分な空きがなければ、領域不足のエラーが発生します。
領域不足に関するエラー番号は、セグメントのタイプごとなどで異なります。次ページで主のエラーを紹介させていただきます。

2. 主な拡張エラー番号についての紹介

表領域内の空き不足によって発生する主なエラーは以下のとおりです。

- ORA-01650 ロールバック・セグメント string を拡張できません (string 分、表領域 string)
- ORA-01650 unable to extend rollback segment %s by %s in tablespace %s
- ORA-01651 アンドウ・セグメントを拡張できません (string 分、表領域 string)
- ORA-01651 unable to extend save undo segment by %s for tablespace %s
- ORA-01652 一時セグメントを拡張できません (string 分、表領域 string)
- ORA-01652 unable to extend temp segment by %s in tablespace %s
- ORA-01653 表 string.string を拡張できません (string 分、表領域 string)
- ORA-01653 unable to extend table %s.%s by %s in tablespace %s
- ORA-01654 索引 string.string を拡張できません (string 分、表領域 string)
- ORA-01654 unable to extend index %s.%s by %s in tablespace %s
- ORA-01655 クラスタ string.string を拡張できません (string 分、表領域 string)
- ORA-01655 unable to extend cluster %s.%s by %s in tablespace %s
- ORA-01688 表: %s.%s のパーティション: %s を拡張できません(%s 分、表領域: %s)
- ORA-01688 unable to extend table %s.%s partition %s by %s in tablespace %s

「string 分」もしくは「%s 分」には、実際に足りなかったデータブロックの数が表示されます。実際のサイズを計算する場合、この数字に初期化パラメータ DB_BLOCK_SIZE を乗じて下さい。以下の例をご覧ください。

ORA-01653: 表 SCOTT.DEPT を拡張できません(256 分、表領域 TOOLS)。

データブロック (db_block_size) の値が 4K の場合、1M 分の連続した空きが TOOLS 表領域に足りなかったためにエラーが発生したということが分かります。

$(256 * 4k) = 1024K$ といった計算式になります。

3. 拡張エラーに対する対処

領域不足によって拡張エラーが発生した場合の対処は、基本的に表領域(データファイル)のサイズを大きくします。

ファイルサイズの拡張には以下の3通りの方法があります。

- 1) データファイルを追加
- 2) データファイルのリサイズ
- 3) データファイルの自動拡張の設定

詳細を説明します。

- 1) データファイルを追加

該当表領域にデータファイルを追加します。

```
ALTER TABLESPACE <表領域名>  
ADD DATAFILE '<ファイルパス>' SIZE n <K|M>;
```

- 2) データファイルのリサイズ

該当表領域に属するデータファイルのサイズを大きくします。

```
ALTER DATABASE DATAFILE '<ファイルパス>'  
RESIZE n <K|M>;
```

- 3) データファイルの自動拡張の設定

該当表領域に属するデータファイルを autoextend on にします。

```
ALTER DATABASE DATAFILE '<ファイルパス>'  
AUTOEXTEND ON;
```

なお、表領域がディクショナリ管理の場合は空き領域の結合 (COALESS) によっても領域不足のエラーを回避できる場合があります。

空き領域を結合できるかどうかは、DBA_FREE_SPACE を検索することで確認できます。
以下の例をご覧ください。

```
SELECT tablespace_name, file_id, block_id, blocks
FROM   dba_free_space
WHERE  tablespace_name = 'USERS'
ORDER BY file_id, block_id;
```

出力例)

TABLESPACE_NAME	FILE_ID	BLOCK_ID	BLOCKS	
TEST	7	545	35	<=
TEST	7	580	40	<=
TEST	7	1245	25	<=
TEST	7	2230	20	<=

と の空き断片が隣接しているかどうかを調べます。 はブロック番号 545 番から 35 ブロック分が空いています。545 + 35 は、580 になります。

の空き断片を確認しますと、ブロック番号 580 番から始まっています。従って、 と は隣接した空き領域であり、結合可能であることが確認できます。

逆に、「BLOCK_ID + BLOCKS」の結果と、次の行の BLOCK_ID の値が同じ値でなければ隣接した空き領域ではありません。

空き領域を結合することによって必要なサイズの空き領域を作成することが出来れば、エラーは回避出来ます。

その際の SQL は以下のものになります。

```
ALTER TABLESPACE <表領域名> COALESCE;
```

ディクショナリ管理とローカル管理の表領域の違い、ローカル管理の表領域の利点、空き領域の結合の詳細は 12 ページに記載します。

また、セグメントが表や索引の場合ですと、空き領域が十分にある表領域へのセグメントの移行によっても対処できます。

表の場合は以下の構文で表の再作成が可能です。

```
ALTER TABLE <表名> MOVE TABLESPACE <表領域名>;
```

ALTER TABLE <表名> MOVE 文は、バージョン 8.1.5 からの新機能となります。それより前のバージョンでは EXP/IMP、CREATE TABLE ~ AS SELECT 文などで再作成を行います。

索引の場合は以下の構文で再作成が可能です。

```
alter index <索引名> rebuild TABLESPACE <表領域名>;
```

4. 領域管理に関する FAQ

領域管理においてよく頂く質問を元にした FAQ 紹介します。

- 1) OEM から確認した領域使用状況に関する質問
- 2) ローカル管理の一時表領域の使用状況に関する質問
- 3) データファイルのサイズに関する質問

- 1) OEM から確認した領域使用状況に関する質問

● 質問

OEM で表領域内の使用状況を確認しますと、空き領域は多く残っています。にも関わらず、セグメントの拡張エラーが発生してしまいます。

名前	タイプ	エクステント管理	サイズ(MB)	使用量(MB)	使用率 %
CWMLITE	PERMANENT	LOCAL	20,000	0.063	0.31
INDX	PERMANENT	LOCAL	25,000	0.063	0.25
SYSTEM	PERMANENT	DICTIONARY	325,000	190,773	58.70
TEST	PERMANENT	DICTIONARY	1,000	0.082	0.20
TOOLS	PERMANENT	LOCAL	30,000	29,063	96.88
UNDOTBS	UNDO	LOCAL	1,450,000	4,375	0.30
USERS	PERMANENT	LOCAL	660,000	213,438	32.34
DRSYS	PERMANENT	LOCAL	0,000	0.000	0.00
TEMP	TEMPORARY	LOCAL	83,000	82,000	98.80

● 回答

OEM で確認できる領域の使用状況は表領域内の合計の空き領域を表示しておりません。2 ページ目の注釈で説明したように、エクステントを獲得するには連続した空き領域が必要となります。

DBA_FREE_SPACE ビューもしくは 2 ページ目の SELECT 文にてエラーメッセージの不足したブロック数が連続した空き領域として存在していないことが確認できます。

2) ローカル管理の一時表領域の使用状況に関する質問

● 質問

一時表領域の使用率を OEM で確認すると、100%近くになってしまっています。また、DBA_FREE_SPACE ビューの空き状況を確認すると、一時表領域はほとんど空きがない状態になっています。領域不足等の問題が発生する可能性はありますか？

● 回答

表領域の領域(エクステント)管理方法には、ローカル管理とディクショナリ管理の2種類があります。ローカル管理の一時表領域を使用している場合、一時表領域に格納されている一時セグメントのエクステントは、ディクショナリ管理の一時表領域と異なり、データベースを再起動しても解放されない仕様になっております。

理由は、一時エクステント解放によるデータベース起動直後のCPUの消費を避けるためです。このため、一度大規模なディスクソートが行われますと、使用率はずっと高い状況を維持したままになります。

これは正しい動作ですので、特に問題はありません。

以下のSQL文で、一時セグメント内の実際の領域の使用量を確認できます。

```
SELECT tablespace_name, extent_size, total_extents,
       used_extents, free_extents
FROM   v$sort_segment;
```

TABLESPACE	EXTENT_SIZE	TOTAL_EXTENTS	USED_EXTENTS	FREE_EXTENTS
TEMP	128	97	0	97

主な列の意味は以下のとおりです。

- extent_size : 1つのエクステントのサイズ(ブロック数)
- total_extents: その一時セグメントに割り当てられているエクステントの数
- used_extents : 現在セッションによって使用されているエクステントの数
- free_extents : 現在どのセッションにも使われていないフリーなエクステントの数

TOTAL_EXTENTS と FREE_EXTENTS の値が 97 ですので、一時セグメントには 97 個のエクステントが割り当てられていますが、全て使われていない状態(再利用可能な状態)であることが分ります。

従いまして、一時セグメントの拡張エラーなどの問題は発生しにくいと判断できます。

3) データファイルのサイズに関する質問

- 質問

データファイルのサイズが大きければ、拡張エラーが発生せず問題ないように思えます。ファイルサイズが非常に大きい場合、何か問題がありますか。

- 回答

OS コマンドでバックアップ/リストアを行う場合、作業に要する時間はファイルサイズに依存します。例えばファイルサイズが 50M でその中の 10M 分だけが使用されている場合でも、バックアップ時には 50M 分の時間を必要とします。このため、必要以上にファイルサイズが大きいと無駄にバックアップ/リストアに時間を要することになります。

5. ローカル管理の表領域とディクショナリ管理の表領域について

セグメント内のエクステントサイズは表領域のタイプに依存します。バージョン 8.1.5 から、ディクショナリ管理の表領域だけでなくローカル管理の表領域も使用できるようになりました。

ディクショナリ管理が表領域内のエクステントの獲得/開放といった情報をデータディクショナリで管理するのに対し、ローカル管理表領域は表領域内のエクステントのメンテナンスを各データファイル内のビットマップによって管理します。この利点として主に以下のことが挙げられます。

- 1) 空き領域の結合という管理者の負荷の軽減
エクステントのローカル管理は、隣り合った連続空き領域を自動的に確認するので、空きエクステントの結合 (COALESCE) をする必要がありません。
- 2) パフォーマンスの向上
ローカル管理表領域はデータディクショナリに空き領域の記録をしないので、データディクショナリに対するアクセスの競合を減らすことができます。
また、ローカル管理表領域では、ディクショナリ管理表領域で発生する「エクステントを獲得/開放することで、ロールバック・セグメントやデータディクショナリの領域も新たに獲得/開放する」といった再帰的な領域管理を避けることができます。
- 3) 領域管理の簡素化
ローカルに管理されるエクステントのサイズが Oracle によって自動的に決定されるようになります。もしくは、すべてのエクステントを同じサイズにすることもできます。
ディクショナリ管理のように STORAGE で各エクステントサイズを定義する必要はありません。

ここで 1)に関連します空き領域の再利用の仕組みについて説明させていただきます。

例えば、ディクショナリ管理の表領域に 4 つのエクステントから構成されている表があったとします。今回、4 つのエクステントが物理的に連続して存在していたとします。この表を DROP すると、再利用可能な領域に変わります。

つまり、4 つのエクステントサイズ分が空き領域となります。この時 4 つの空き領域の断片が隣接して存在することになるだけで、空き領域は連続した 1 つの空き領域とはみなされません。

この状態では、削除したエクステント 2 つ分のサイズを必要とするような大きなエクステントを割り当てることができません。Oracle は別の場所にエクステントを割り当ててしまい、隣接している 4 つの空き領域はそのまま空き断片として残ってしまいます。

このように、連続する小さな空き領域を合わせて 1 つの大きな空き領域にしないと、領域を効率よく使用することができません。そこで、COALESCE(隣接した空き領域の結合)の作業が必要になります。

最後に

今回は、表領域内の使用状況を確認するための SQL や、拡張エラーに対する対処などをご紹介させて頂きました。今回のスクリプトや対処法が皆様のお役に立てると幸いです。

参考資料

Oracle9i データベース管理者ガイド Vol.1 リリース 1 (9.0.1)
Oracle9iSQL リファレンス Vol.1 リリース 1 (9.0.1)
Oracle9i データベース・エラー・メッセージリリース 1 (9.0.1)