2018年8月17日

11:53

**检查及设置合理的undo表空间**

2013年10月24日 15:39:58

阅读数：4610

      UNDO是用于实现并发控制以及构建一致性读，也就是在数据变更之前产生前镜像，以保证用户能够回滚或撤销对数据库所作的修改。是Oracle数据库完整性的重要组成部分。因此合理的设计及配置undo以及使用undo都将对数据库有较大的影响。通常情况下，对于大规模数据的删除，更新操作，我们建议使用分批删除分次提交以减少对undo的占用和冲击。那么对于undo的大小到底应该设置多大？是启用自动扩展还是关闭自动扩展？这个问题仁者见仁，智者见智，见下文。

1、当前数据库环境及undo配置信息

* 1. sys@SYTST> select \* from v$version where rownum<2;
  2. BANNER
  3. *----------------------------------------------------------------*
  4. Oracle Database 10g Release 10.2.0.3.0 - 64bit Production
  5. sys@SYTST> show parameter undo
  6. NAME TYPE VALUE
  7. *------------------------------------ ----------- ------------------------------*
  8. undo\_management string AUTO
  9. undo\_retention integer 900
  10. undo\_tablespace string UNDOTBS
  11. sys@SYTST> select tablespace\_name,file\_name,AUTOEXTENSIBLE,bytes/1024/1024 size\_mb
  12. 2 from dba\_data\_files where tablespace\_name like '%UNDO%';
  13. TABLESPACE\_NAME FILE\_NAME AUT SIZE\_MB
  14. *------------------------------ ------------------------------------------------------- --- ----------*
  15. UNDOTBS /u02/database/SYTST/undo/undoSYTST1.dbf NO 20
  16. *--创建演示表t*
  17. sys@SYTST> CREATE TABLE t
  18. 2 AS
  19. 3 SELECT rownum AS id,
  20. 4 round(5678+dbms\_random.normal\*1234) AS n1,
  21. 5 mod(255+trunc(dbms\_random.normal\*1000),255) AS n2,
  22. 6 dbms\_random.string('p',255) AS pad
  23. 7 FROM dual
  24. 8 CONNECT BY level <= 100000
  25. 9 ORDER BY dbms\_random.value;
  26. Table created.
  27. *--直接用脚本得到undo的信息及建议值*
  28. sys@SYTST> @chk\_advs\_undo
  29. Session altered.
  30. - Undo Analysis started at : 24/10/2013 14:39:58 -
  31. *--------------------------------------------------*
  32. NOTE:The following analysis is based upon the database workload during the period -
  33. Begin Time : 17/10/2013 14:39:58
  34. End Time : 24/10/2013 14:39:58
  35. Current Undo Configuration
  36. *--------------------------*
  37. Current undo tablespace : UNDOTBS
  38. Current undo tablespace size (datafile size now) : 20M
  39. Current undo tablespace size (consider autoextend) : 20M
  40. AUTOEXTEND for undo tablespace is : OFF
  41. Current undo retention : 900
  42. UNDO GUARANTEE is set to : FALSE
  43. Undo Advisor Summary
  44. *---------------------------*
  45. Finding 1:The undo tablespace is OK. *--->当前的undo配置合理*
  46. Undo Space Recommendation
  47. *-------------------------*
  48. Allocated undo space is sufficient for the current workload.
  49. Retention Recommendation
  50. *------------------------*
  51. The best possible retention with current configuration is : 5996 Seconds
  52. The longest running query ran for : 52 Seconds
  53. The undo retention required to avoid errors is : 52 Seconds
  54. PL/SQL procedure successfully completed.

2、模拟undo超出并获得建议值

* 1. *--先查看当前已产生的undo*
  2. sys@SYTST> @mystat "undo change"
  3. sys@SYTST> set echo off
  4. NAME VALUE
  5. ---------------------------------------------------------------- ----------
  6. undo change vector size 363568
  7. --我们来更新之前创建表t上的pad列
  8. sys@SYTST> update t set pad=dbms\_random.string('l',255);
  9. update t set pad=dbms\_random.string('l',255)
  10. \*
  11. ERROR at line 1:
  12. ORA-30036: unable to extend segment by 8 in undo tablespace 'UNDOTBS' --->得到了错误提示，无法扩展undo，因为当前环境undo未启用自动扩展
  13. --Author : Leshami
  14. --Blog : <http://blog.csdn.net/leshami>
  15. --看看update语句到底产生了多少undo
  16. sys@SYTST> @mystat2
  17. sys@SYTST> set echo off
  18. NAME V DIFF
  19. ---------------------------------------------------------------- ---------- ----------------
  20. undo change vector size 18817316 18,453,748
  21. --由下可知，上面的update语句已经产生了17M以上的undo,而当前的undo的大小为20M，显然不够，所以我们收到了ORA-30036
  22. sys@SYTST> select 18453748/1024/1024 from dual;
  23. 18453748/1024/1024
  24. ------------------
  25. 17.5988655
  26. --再次获得undo建议值
  27. sys@SYTST> @chk\_advs\_undo
  28. Session altered.
  29. - Undo Analysis started at : 24/10/2013 14:49:07 -
  30. --------------------------------------------------
  31. NOTE:The following analysis is based upon the database workload during the period -
  32. Begin Time : 17/10/2013 14:49:07
  33. End Time : 24/10/2013 14:49:07
  34. Current Undo Configuration
  35. --------------------------
  36. Current undo tablespace : UNDOTBS
  37. Current undo tablespace size (datafile size now) : 20M
  38. Current undo tablespace size (consider autoextend) : 20M
  39. AUTOEXTEND for undo tablespace is : OFF
  40. Current undo retention : 900
  41. UNDO GUARANTEE is set to : FALSE
  42. Undo Advisor Summary
  43. ---------------------------
  44. Finding 1:Undo Tablespace is under pressure. Recommendation 1:Size undo tablespace to 48 MB
  45. Undo Space Recommendation
  46. -------------------------
  47. Minimum Recommendation : Size undo tablespace to 48 MB
  48. Rationale : Increase undo tablespace size so that long running queries will not fail
  49. Recommended Undo Tablespace Size : 48M --->这里给出了undo的建议值
  50. Retention Recommendation
  51. ------------------------ --->下面是一些和undo相关的建议，如可能的保留时间等等
  52. The best possible retention with current configuration is : 306 Seconds
  53. The longest running query ran for : 52 Seconds
  54. The undo retention required to avoid errors is : 52 Seconds
  55. PL/SQL procedure successfully completed.

3、获得undo信息及建议值脚本

* 1. *--脚本来自Oracle. 参考Metalink: Doc ID 1579035.1*
  2. *--以下脚本适用于Oracle 10g以上版本*
  3. SET SERVEROUTPUT ON
  4. SET LINES 600
  5. ALTER SESSION SET NLS\_DATE\_FORMAT = 'DD/MM/YYYY HH24:MI:SS';
  6. DECLARE
  7. v\_analyse\_start\_time DATE := SYSDATE - 7;
  8. v\_analyse\_end\_time DATE := SYSDATE;
  9. v\_cur\_dt DATE;
  10. v\_undo\_info\_ret BOOLEAN;
  11. v\_cur\_undo\_mb NUMBER;
  12. v\_undo\_tbs\_name VARCHAR2(100);
  13. v\_undo\_tbs\_size NUMBER;
  14. v\_undo\_autoext BOOLEAN;
  15. v\_undo\_retention NUMBER(6);
  16. v\_undo\_guarantee BOOLEAN;
  17. v\_instance\_number NUMBER;
  18. v\_undo\_advisor\_advice VARCHAR2(100);
  19. v\_undo\_health\_ret NUMBER;
  20. v\_problem VARCHAR2(1000);
  21. v\_recommendation VARCHAR2(1000);
  22. v\_rationale VARCHAR2(1000);
  23. v\_retention NUMBER;
  24. v\_utbsize NUMBER;
  25. v\_best\_retention NUMBER;
  26. v\_longest\_query NUMBER;
  27. v\_required\_retention NUMBER;
  28. BEGIN
  29. select sysdate into v\_cur\_dt from dual;
  30. DBMS\_OUTPUT.PUT\_LINE(CHR(9));
  31. DBMS\_OUTPUT.PUT\_LINE('- Undo Analysis started at : ' || v\_cur\_dt || ' -');
  32. DBMS\_OUTPUT.PUT\_LINE('*--------------------------------------------------');*
  33. v\_undo\_info\_ret := DBMS\_UNDO\_ADV.UNDO\_INFO(v\_undo\_tbs\_name, v\_undo\_tbs\_size, v\_undo\_autoext, v\_undo\_retention, v\_undo\_guarantee);
  34. select sum(bytes)/1024/1024 into v\_cur\_undo\_mb from dba\_data\_files where tablespace\_name = v\_undo\_tbs\_name;
  35. DBMS\_OUTPUT.PUT\_LINE('NOTE:The following analysis is based upon the database workload during the period -');
  36. DBMS\_OUTPUT.PUT\_LINE('Begin Time : ' || v\_analyse\_start\_time);
  37. DBMS\_OUTPUT.PUT\_LINE('End Time : ' || v\_analyse\_end\_time);
  38. DBMS\_OUTPUT.PUT\_LINE(CHR(9));
  39. DBMS\_OUTPUT.PUT\_LINE('Current Undo Configuration');
  40. DBMS\_OUTPUT.PUT\_LINE('*--------------------------');*
  41. DBMS\_OUTPUT.PUT\_LINE(RPAD('Current undo tablespace',55) || ' : ' || v\_undo\_tbs\_name);
  42. DBMS\_OUTPUT.PUT\_LINE(RPAD('Current undo tablespace size (datafile size now) ',55) || ' : ' || v\_cur\_undo\_mb || 'M');
  43. DBMS\_OUTPUT.PUT\_LINE(RPAD('Current undo tablespace size (consider autoextend) ',55) || ' : ' || v\_undo\_tbs\_size || 'M');
  44. IF V\_UNDO\_AUTOEXT THEN
  45. DBMS\_OUTPUT.PUT\_LINE(RPAD('AUTOEXTEND for undo tablespace is',55) || ' : ON');
  46. ELSE
  47. DBMS\_OUTPUT.PUT\_LINE(RPAD('AUTOEXTEND for undo tablespace is',55) || ' : OFF');
  48. END IF;
  49. DBMS\_OUTPUT.PUT\_LINE(RPAD('Current undo retention',55) || ' : ' || v\_undo\_retention);
  50. IF v\_undo\_guarantee THEN
  51. DBMS\_OUTPUT.PUT\_LINE(RPAD('UNDO GUARANTEE is set to',55) || ' : TRUE');
  52. ELSE
  53. dbms\_output.put\_line(RPAD('UNDO GUARANTEE is set to',55) || ' : FALSE');
  54. END IF;
  55. DBMS\_OUTPUT.PUT\_LINE(CHR(9));
  56. SELECT instance\_number INTO v\_instance\_number FROM V$INSTANCE;
  57. DBMS\_OUTPUT.PUT\_LINE('Undo Advisor Summary');
  58. DBMS\_OUTPUT.PUT\_LINE('*---------------------------');*
  59. v\_undo\_advisor\_advice := dbms\_undo\_adv.undo\_advisor(v\_analyse\_start\_time, v\_analyse\_end\_time, v\_instance\_number);
  60. DBMS\_OUTPUT.PUT\_LINE(v\_undo\_advisor\_advice);
  61. DBMS\_OUTPUT.PUT\_LINE(CHR(9));
  62. DBMS\_OUTPUT.PUT\_LINE('Undo Space Recommendation');
  63. DBMS\_OUTPUT.PUT\_LINE('*-------------------------');*
  64. v\_undo\_health\_ret := dbms\_undo\_adv.undo\_health(v\_analyse\_start\_time, v\_analyse\_end\_time, v\_problem, v\_recommendation, v\_rationale, v\_retention, v\_utbsize);
  65. IF v\_undo\_health\_ret > 0 THEN
  66. DBMS\_OUTPUT.PUT\_LINE('Minimum Recommendation : ' || v\_recommendation);
  67. DBMS\_OUTPUT.PUT\_LINE('Rationale : ' || v\_rationale);
  68. DBMS\_OUTPUT.PUT\_LINE('Recommended Undo Tablespace Size : ' || v\_utbsize || 'M');
  69. ELSE
  70. DBMS\_OUTPUT.PUT\_LINE('Allocated undo space is sufficient for the current workload.');
  71. END IF;
  72. SELECT dbms\_undo\_adv.best\_possible\_retention(v\_analyse\_start\_time, v\_analyse\_end\_time) into v\_best\_retention FROM dual;
  73. SELECT dbms\_undo\_adv.longest\_query(v\_analyse\_start\_time, v\_analyse\_end\_time) into v\_longest\_query FROM dual;
  74. SELECT dbms\_undo\_adv.required\_retention(v\_analyse\_start\_time, v\_analyse\_end\_time) into v\_required\_retention FROM dual;
  75. DBMS\_OUTPUT.PUT\_LINE(CHR(9));
  76. DBMS\_OUTPUT.PUT\_LINE('Retention Recommendation');
  77. DBMS\_OUTPUT.PUT\_LINE('*------------------------');*
  78. DBMS\_OUTPUT.PUT\_LINE(RPAD('The best possible retention with current configuration is ',60) || ' : ' || v\_best\_retention || ' Seconds');
  79. DBMS\_OUTPUT.PUT\_LINE(RPAD('The longest running query ran for ',60) || ' : ' || v\_longest\_query || ' Seconds');
  80. DBMS\_OUTPUT.PUT\_LINE(RPAD('The undo retention required to avoid errors is ',60) || ' : ' || v\_required\_retention || ' Seconds');
  81. END;
  82. /

4、后记

a、undo究竟要多大，推荐的做法是根据你的业务高峰期得到一个峰值，然后适当高于峰值来启用自动扩展。

b、undo保留时间与所需的空间成正比，保留时间越长，空间开销越大，不过出现ORA-01555的几率也就越低。

c、在未启用自动扩展的情形下，对于新的事务，UNDO\_RETENTION会回收空闲空间，并重新使用，如果启用自动扩展，则该功能被忽略，而是先扩展。

d、小的undo表空间(数据文件)通常情况下对于数据库及操作系统影响更小，具有更好的性能以及减少备份所需的开销等。

源文档 <<https://blog.csdn.net/leshami/article/details/12999581>>