

Skillset2

Skillset2

解决oms无法登录的问题:

这个题目是北京那边崔老师的环境的问题, 我们上海这边的环境没有这样的问题, 考试也没有这样的问题, 所以不需要我们进行考虑

Section 1: 物化视图

1.配置PDBPROD1数据库, 使其中的物化视图支持查询重写, 即使在约束没有被验证的情况下

```
SQL> alter system set query_rewrite_enabled=true;  
SQL> alter system set query_rewrite_integrity=trusted;
```

Section 2: 表空间传输

1.使用表空间传输方法, 将TRPDATA传输到PDBPROD1数据库中。

```
set linesize 250  
select * from v$transportable_platform;  
convert datafile '/home/oracle/scripts/transport_tbs01.dbf' from platform 'Solaris[tm] OE (32-bit)' format '/tmp/test01.dbf';  
  
alter session set container=pdbprod1;  
create user trans_user identified by oracle;  
  
drop directory kiwi;  
create directory kiwi as '/home/oracle/scripts';  
grant read,write on directory kiwi to public;  
impdp system/oracle@pdbprod1 dumpfile=TRPDATA.DMP directory=kiwi transport_datafiles=/tmp/test01.dbf
```

Section 3: 优化星型查询

1.配置PDBPROD1数据库支持星型查询, 确保这个改变不影响其他PRODCDB中的其他PDB

```
alter system set star_transformation_enabled=true;
```

.

Section 4: 配置并行执行

1.在PDBPROD1中配置并行执行, 使用SYS用户, 执行如下:

(1)设置sh.sales表的并行度为默认并行度

```
alter table sh.sales parallel;
```

(2)配置PDBPROD1使用自动并行度, 只有当语句中的对象没有在数据字典中指定并行度时生效。

```
alter system set parallel_degree_policy=limited;
```

Section 5: 管理分区

1.在PDBPROD1中, SH用户下创建间隔分区

表名:sales_history_2014,按照2011, 2012, 2013, 2014年份来分区

分区名为SAL1, SAL2, SAL3, SAL4

基于time_id分区

列

PROD_ID number not null

```
CUST_ID number not null
TIME_ID DATE not null
CHANNEL_ID number not null
PROMO_ID number not null
QUANTITY_SOLD number(10,2) not null
AMOUNT_SOLD number(10,2) not null
```

```
create table sales_history_2014
(
    PROD_ID number not null,
    CUST_ID number not null,
    TIME_ID DATE not null,
    CHANNEL_ID number not null,
    PROMO_ID number not null,
    QUANTITY_SOLD number(10,2) not null,
    AMOUNT_SOLD number(10,2) not null
    partition by range (time_id)
    interval (numtoyminterval(1,'year'))
    (partition sal1 values less than (to_date('01-01-2012','dd-mm-yyyy')),
    partition sal2 values less than (to_date('01-01-2013','dd-mm-yyyy')),
    partition sal3 values less than (to_date('01-01-2014','dd-mm-yyyy')),
    partition sal4 values less than (to_date('01-01-2015','dd-mm-yyyy'))
);
```

使用脚本加载数据

s2_1.sql

```
insert into sh.sales_history_2014 values(115,10795,to_date('23-FEB-12','DD-MON-RR'),3,999,1, 9.3);
insert into sh.sales_history_2014 values(115, 1798,to_date('23-FEB-12','DD-MON-RR'),4,999,1, 9.3);
insert into sh.sales_history_2014 values(115, 9041,to_date('23-FEB-12','DD-MON-RR'),4,999,1, 9.3);
insert into sh.sales_history_2014 values(115, 1321,to_date('24-FEB-13','DD-MON-RR'),2,999,1, 9.17);
insert into sh.sales_history_2014 values(115, 1321,to_date('24-FEB-13','DD-MON-RR'),3,999,1, 9.17);
insert into sh.sales_history_2014 values(115, 2088,to_date('24-FEB-13','DD-MON-RR'),3,999,1, 9.17);
insert into sh.sales_history_2014 values(115, 2357,to_date('24-FEB-14','DD-MON-RR'),3,999,1, 9.17);
insert into sh.sales_history_2014 values(115, 6097,to_date('24-FEB-14','DD-MON-RR'),3,999,1, 9.17);
insert into sh.sales_history_2014 values(115, 7526,to_date('24-FEB-14','DD-MON-RR'),3,999,1, 9.17);
insert into sh.sales_history_2014 values(115,11817,to_date('24-FEB-11','DD-MON-RR'),3,999,1, 9.17);
insert into sh.sales_history_2014 values(115,25121,to_date('24-FEB-11','DD-MON-RR'),3,999,1, 9.17);
insert into sh.sales_history_2014 values(116, 465,to_date('01-JAN-11','DD-MON-RR'),3,999,1, 12.4);
insert into sh.sales_history_2014 values(116, 5345,to_date('01-JAN-12','DD-MON-RR'),3,999,1, 12.4);
insert into sh.sales_history_2014 values(116,12149,to_date('01-JAN-12','DD-MON-RR'),3,999,1, 12.4);
insert into sh.sales_history_2014 values(116, 5345,to_date('01-JAN-12','DD-MON-RR'),4,999,1, 12.4);
insert into sh.sales_history_2014 values(116, 819,to_date('02-JAN-13','DD-MON-RR'),3,999,1,14.59);
insert into sh.sales_history_2014 values(116, 1562,to_date('02-JAN-13','DD-MON-RR'),3,999,1,14.59);
insert into sh.sales_history_2014 values(116, 3581,to_date('02-JAN-13','DD-MON-RR'),3,999,1,14.59);
insert into sh.sales_history_2014 values(116, 6485,to_date('02-JAN-13','DD-MON-RR'),3,999,1,14.59);
insert into sh.sales_history_2014 values(116, 6683,to_date('02-JAN-14','DD-MON-RR'),3,999,1,14.59);
insert into sh.sales_history_2014 values(116, 547,to_date('04-JAN-14','DD-MON-RR'),3,999,1, 12.4);
insert into sh.sales_history_2014 values(116, 1369,to_date('04-JAN-14','DD-MON-RR'),3,999,1, 12.4);
```

Section 6: 配置闪回技术

1.配置PDBPROD1, 实现闪回归档

表空间名称为:FLASH_ARC, 数据文件 50M

```
create tablespace FLASH_ARC datafile '/u01/app/oracle/oradata/PRODCDB/pdbprod1/FLASH_ARC01.dbf' size 50m autoextend on;  
用户名为:FARC_OWNER
```

```
create user FARC_OWNER identified by oracle;  
alter user FARC_OWNER default tablespace FLASH_ARC quota unlimited on FLASH_ARC;  
grant flashback archive administer to FARC_OWNER;  
grant dba to FARC_OWNER;  
create flashback archive flasharc1 tablespace FLASH_ARC retention 1 year;  
alter table SH.PROMOTIONS flashback archive flasharc1;
```

SH.PROMOTIONS表之前的版本, 能够通过flasharc1 闪回归档查询到

Section 7: 使用EM Cloud Control 配置资源管理器

EM CC安装在host02中。在host01上安装agent, 发现host01和host02

这部分我们使用sqldeveloper来进行操作, 因为在cc中, 不能够完成所有的操作

为PRODCDB配置资源管理器

1.分配SYSTEM用户作为资源管理员权限

2.为PRODCDB创建资源计划为DAYTIMEP

PDBPROD1得到6份共享CDB资源

PDBPROD2得到3份共享CDB资源

PDBPROD3得到1份共享CDB资源

没有PDB可以得到多余75%的可用cpu时间

没有PDB可以得到多余50%的parallel_servers_target

3.在PDBPROD1中创建资源消耗组 ONLINE_GRP和REPORTING_GRP

使用注释标明该组用途

4.在PDBPROD1中创建资源管理计划

(1)计划名称:DAYTIMEC

(2)对于ONLINE_GRP组, 不能允许超过25个活动会话。如果26个用户尝试登录, 等待90秒以后, 这个需求将被终止。

(3)REPORTING_GRP组最大活动会话数量为8.如果需要多余8个会话150秒以后这个要求将被终止。

(4)在ONLINE_GRP组, 一个会话最长查询执行时间为5秒, 如果查询估算超过5s, 则查询应该报错

(5)ONLINE_GRP组最大undo为350M

(6)cpu比率, ONLINE_GRP, REPORTING_GRP, OTHER_GROUPS分别为50, 40, 10

(7)REPORTING_GRP组应该限制并行度为30

(8) 如果会话超过8000i/o或者超过2000m的数据传输, 任何在REPORTING_grp组都应该被临时交互到low_group组中。当调用完成, 会话还回到组中。

5 daytimep 激活, daytimec也要激活

Section 8: 调优sql语句

1.使用result cache

在PDBPROD1数据库中, 设置result cache size 15m,

, 让脚本中的语句都可以使用result cache

这个在cdb层面进行操作

```
alter system set result_cache_max_size=15m;
```

这个在pdbprod1里面改

```
alter system set result_cache_mode=force;
```

这儿要考察使用result cache, 我们看脚本中有update的语句, 我们知道, 有表中只要数据有了变化, 那么result cache中的对象就会失效, 变为invalid的状态

所以, 原来我们采取的是修改脚本添加hint的做法, 但想想开始不够, 这儿我们是直接修改参数, 来强制使用result cache的做法

2.多列统计信息

登录PDBPROD1数据库中的SH用户, 运行脚本, 脚本中的查询, CUST_CITY, CUST_STATE_PROVINCE,COUNTRY_ID列经常一起使用。提高优化器对这些语句计算的可选择率

```
select dbms_stats.create_extended_stats('SH','TEST','(CUST_CITY,CUST_STATE_PROVINCE,COUNTRY_ID)') from dual;  
exec dbms_stats.gather_table_stats('SH','TEST',CASCADE=>TRUE,METHOD_OPT=>'FOR ALL COLUMNS SIZE AUTO FOR COLUMNS(CUST_CITY,CUST_STATE_PROVINCE,COUNTRY_ID) SIZE AUTO');
```

```
cd /u01/app/oracle/product/weblogic/oms/sysman/prov/agentpush
```

```
[oracle@enmoedu2 agentpush]$ pwd
```

```
vi /u01/app/oracle/product/weblogic/oms/sysman/prov/agentpush/agentpush.properties
```

```
date -s "10:15:24 Jan 3, 2016"
```

```
Sun Jan 3 13:36:34 CST 2016
```

```
oracle.sysman.prov.agentpush.enablePty=true
```

```
[oracle@enmoedu2 agentpush]$ ls
```

```
agentpush.properties jlib scripts xmls
```

```
oracle.sysman.prov.agentpush.enablePty=true
```

3.

Dump file location for: /home/oracle/scripts,the file :appsandstage.dmp.

```
impdp system/oracle@pdbprod1 dumpfile=appsandstage.dmp directory=kiwi;
```

```
exec dbms_sqltune.UNPACK_STGTAB_SQLSET('%','% ',true,'STS_JFV_PS','APPS');
```

Migration from 11.2.0.3 to 12.1.0.2

4.SPM, 演进基线

在PDBPROD1的SH用户下, 执行脚本, 并演进基线

Section 9: 索引调优

1.在向PDBPROD1数据库sh用户的PROD_INFO表插入数据时, 发现大量的buffer busy waits 等待事件。PRODUCT_ID列上一个序列被使用。

范围扫描也频繁执行在PRODUCT_ID列上。创建一个索引PROD_IDX, 在PRODUCT_ID列上, 不要产生大量的buffer buffer waits等待, 并且可以用于范围扫描。

创建全局hash 分区索引

```
create table sh.prod_info as select * from oe.product_information;
```

```
create index prod_idx on prod_info (product_id) global partition by hash(product_id) partitions 4 parallel 4;
```