

# ORACLE ADVANCED PRODUCT SERVICES

中国移动通信集团公司

流程平台上线压测表改造

V1.0

ORACLE工程师： 刘晓建

报告生成日期： 2018年09月13日

现场支持起始日期： 2018年09月13日

现场支持结束日期： 2018年09月14日

现场支持总时间（小时）：

Timesheet编号：



## 文档控制

此文档仅供中国移动通信集团公司与Oracle公司支持服务部审阅

不得向与此无关的个人或机构传阅或复制

### 修改记录

日期	作者	版本	修改记录
2018-09-13	刘晓建	1.0	创建初始版本

### 审阅记录

日期	审阅人	职位
2019-09-13	胡奇虎	ORACLE 北区ACS高级经理
2018-09-13	刘璐	TAM

### 分发记录

日期	拷贝No.	姓名	单位
2018-09-13		苏伟	压测项目负责人

# 目 录

文档控制 .....	II
1. 本次服务总结 .....	1
2. 表和索引改造内容 .....	1
2.1 数据和索引进行表空间隔离 .....	1
2.1.1 新建存放索引的表空间(2套库均已完成).....	1
2.1.2 查看当前索引存放所用表空间.....	2
2.1.3 处理NORMAL类型的索引.....	3
2.1.4 处理LOB 类型的索引 .....	3
2.1.5 处理状态无效的索引.....	4
2.2 单表改为分区表 .....	4
2.2.1 验证是否能进行改造.....	4
2.2.2 收集要改造表的定义.....	4
2.2.3 创建中间表（按天自动分区） .....	7
2.2.4 在中间表上创建索引.....	9
2.2.5 开始重定义.....	10
2.2.6 同步数据.....	10
2.2.7 完成重定义.....	10
2.2.8 验证表定义是否被转为分区表.....	10
2.2.9 确认是否存在状态为unusable的.....	10
2.2.10 删除在线重定义生成的临时表.....	10
2.3 主键索引改为哈希索引 .....	10
2.3.1 先确认要改造的主键索引和约束的名字.....	10
2.3.2 删除已有主键约束.....	11
2.3.3 创建唯一索引.....	12
2.3.4 增加主键约束.....	12
2.3.5 调整各个子分区的存储属性.....	12
2.3.6 --各分区状态确认 .....	12
2.4 2节点数据未完成工作 .....	13
3. 参考命令 .....	13

## 1. 本次服务总结

### 背景

流程系统上线前压测，需要对2套Oracle rac 数据库进行性能优化。

Redhat 7.3 + 128g(mem)+64(逻辑cpu)

12.1.0.2 +4节点rac和2节点rac+归档模式

2TB的磁盘组（外部冗余）

### 优化工作

优化项目	主要内容	是否完成
WFPPRD库USERS表空间对象迁移	将业务用户存储到users表空间的对象分别迁移到TS_WFP和TS_INDX（新建）	完成
数据和索引存放的表空间进行分离	新建索引表空间TS_INDX 将8个业务用户索引迁移到TS_INDX	完成
表空间	大部分的表空间都进行了初步扩容	完成
主键索引改为哈希索引	2个库共13个用户的表WFPROCESSINST 的主键改造	仅WFPPRD01库的为完成，因为一直有业务占用对象，请找时间按照2.4部分进行操作
提供基本维护的命令	提供了详细的操作命令	完成

## 2. 表和索引改造内容

### 2.1 数据和索引进行表空间隔离

#### 2.1.1 新建存放索引的表空间(2 套库均已完成)

```
create tablespace TS_INDX datafile '+DATADG' size 30g;
```

```
alter tablespace TS_INDX datafile '+DATADG' size 30g;
```

```
alter tablespace TS_INDX datafile '+DATADG' size 30g;
```

### 2.1.2 查看当前索引存放所用表空间

```
select owner,index_type,tablespace_name,count(*) from dba_indexes where  
  
owner in (  
  
'RBS',  
  
'ERPPRD',  
  
'ITPRD',  
  
'CMSPRD',  
  
'SOAPRD',  
  
'MDMPRD',  
  
'AAMPRD',  
  
'TAXPRD',  
  
'CMFPRD',  
  
'WFPPRD')  
  
group by owner,index_type,tablespace_name order by 1,2;
```

OWNER	INDEX_TYPE	TABLESPACE_NAME	COUNT(*)
-----			
AAMPRD	LOB	TS_WFP	21
AAMPRD	NORMAL	TS_WFP	203
AAMPRD	NORMAL		3
CMFPRD	LOB	TS_WFP	19
CMFPRD	NORMAL	TS_WFP	204
CMSPRD	LOB	TS_WFP	19
CMSPRD	NORMAL	TS_WFP	203
CMSPRD	NORMAL		1
ERPPRD	LOB	TS_WFP	19
ERPPRD	NORMAL	TS_WFP	204
ITPRD	LOB	TS_WFP	19
ITPRD	NORMAL	TS_WFP	204
MDMPRD	LOB	TS_WFP	19

MDMPRD	NORMAL	TS_WFP	204
SOAPRD	LOB	TS_WFP	19
SOAPRD	NORMAL	TS_WFP	204
TAXPRD	LOB	TS_WFP	19
TAXPRD	NORMAL	TS_WFP	204
WFPPRD	LOB	TS_WFP	19
WFPPRD	LOB	USERS	4
WFPPRD	NORMAL	TS_WFP	208
WFPPRD	NORMAL	USERS	19

### 2.1.3 处理 **NORMAL** 类型的索引

```
select 'alter index ' || owner || '.' || index_name || ' rebuild tablespace TS_INDX;' RBINDX
from dba_indexes where owner in ('RBS',
'ERPPRD',
'ITPRD',
'CMSPRD',
'SOAPRD',
'MDMPRD',
'AAMP RD',
'TAXPRD',
'CMFPRD',
'WFPPRD') and index_type='NORMAL';
```

#将上述结果进行执行

### 2.1.4 处理 **LOB** 类型的索引

LOB 类型的列对应的索引只能通过move 表的方式进行处理

```
select 'alter table ' || OWNER || '.' || TABLE_name || ' move lob(' || column_name || ') ' || ' store as
(tablespace TS_INDX);'
from dba_lobs where tablespace_name in ('TS_WFP','USERS') and
owner in (
'RBS',
'ERPPRD',
```

```
'ITPRD',
'CMSPRD',
'SOAPRD',
'MDMPRD',
'AAMPRD',
'TAXPRD',
'CMFPRD',
'WFPPRD') ;
```

### 2.1.5 处理状态无效的索引

```
SELECT 'alter index ' || owner || '.' || index_name || ' rebuild tablespace ts_idx;'
FROM DBA_INDEXES where status='UNUSABLE' ;
```

将状态无效的索引，进行批量集中处理。

## 2.2 单表改为分区表

通过在线重定义将每个用户的2个表进行按天分区的改造，拿其中一个用户：'CMSPRD' 和表 'WFWIPARTICIPANT'进行举例说明：

### 2.2.1 验证是否能进行改造

```
call
DBMS_REDEFINITION.CAN_REDEF_TABLE(uname=>'CMSPRD',tname=>'WFWIPARTICIPAN
T',options_flag=>1);
```

不返回任何结果说明可以进行重定义。

### 2.2.2 收集要改造表的定义

```
Set long 20000

Select dbms_metadata.get_ddl('TABLE',' WFWIPARTICIPANT',' CMSPRD') from dual;

DBMS_METADATA.GET_DDL('TABLE','WFWIPARTICIPANT', CMSPRD')
-----

CREATE TABLE CMSPRD.WFWIPARTICIPANT
(  WIPARTICID NUMBER NOT NULL ENABLE,
   WORKITEMID NUMBER,
```

PARTICIPANTTYPE VARCHAR2(20),  
PARTICIPANTID VARCHAR2(64),  
PARTICIPANTNAME VARCHAR2(256),  
PARTIINTYPE VARCHAR2(20),  
DELEGATETYPE VARCHAR2(10),  
PARTICIPANTINDEX NUMBER,  
GLOBALID VARCHAR2(64),  
WORKITEMNAME VARCHAR2(256),  
WORKITEMTYPE VARCHAR2(20),  
WORKITEMDESC VARCHAR2(512),  
CURRENTSTATE NUMBER(2,0),  
PARTICIPANT VARCHAR2(128),  
PRIORITY NUMBER(4,0),  
ISTIMEOUT CHAR(1),  
LIMITNUM NUMBER,  
LIMITNUMDESC VARCHAR2(512),  
CREATETIME DATE,  
STARTTIME DATE,  
ENDTIME DATE,  
FINALTIME DATE,  
REMINDTIME DATE,  
ACTIONURL VARCHAR2(256),  
PROCESSINSTID NUMBER,  
ACTIVITYINSTID NUMBER,  
STATESLIST VARCHAR2(256),  
TIMEOUTNUM NUMBER,  
TIMEOUTNUMDESC VARCHAR2(512),  
PROCESSINSTNAME VARCHAR2(256),  
ACTIVITYINSTNAME VARCHAR2(256),  
PROCESSDEFID NUMBER,



```

PROCESSDEFNAME VARCHAR2(256),
PROCESSCHNAME VARCHAR2(256),
ACTIVITYDEFID VARCHAR2(64),
ASSISTANT VARCHAR2(64),
BIZSTATE NUMBER(*,0),
ALLOWAGENT VARCHAR2(10),
ROOTPROCINSTID NUMBER,
ACTIONMASK VARCHAR2(64),
PARTINAME VARCHAR2(64),
ASSISTANTNAME VARCHAR2(64),
URLTYPE VARCHAR2(20),
DEALRESULT VARCHAR2(512),
DEALOPINION VARCHAR2(256),
EXTEND1 VARCHAR2(64),
EXTEND2 VARCHAR2(64),
EXTEND3 VARCHAR2(64),
EXTEND4 VARCHAR2(64),
EXTEND5 NUMBER,
CATALOGUUID VARCHAR2(32),
CATALOGNAME VARCHAR2(64),
TENANT_ID VARCHAR2(128),
PRESSNUM NUMBER(32,0),
CONSTRAINT PK_WFWIPARTICIPANT PRIMARY KEY (WIPARTICID)
USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
TABLESPACE TS_INDX  ENABLE
) SEGMENT CREATION DEFERRED
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
TABLESPACE TS_WFP ;

```

### 2.2.3 创建中间表（按天自动分区）

```
CREATE TABLE CMSPRD.WFWIPARTICIPANT_TEMP01
(
  WIPARTICID NUMBER NOT NULL ENABLE,
  WORKITEMID NUMBER,
  PARTICIPANTTYPE VARCHAR2(20),
  PARTICIPANTID VARCHAR2(64),
  PARTICIPANTNAME VARCHAR2(256),
  PARTIINTYPE VARCHAR2(20),
  DELEGATETYPE VARCHAR2(10),
  PARTICIPANTINDEX NUMBER,
  GLOBALID VARCHAR2(64),
  WORKITEMNAME VARCHAR2(256),
  WORKITEMTYPE VARCHAR2(20),
  WORKITEMDESC VARCHAR2(512),
  CURRENTSTATE NUMBER(2,0),
  PARTICIPANT VARCHAR2(128),
  PRIORITY NUMBER(4,0),
  ISTIMEOUT CHAR(1),
  LIMITNUM NUMBER,
  LIMITNUMDESC VARCHAR2(512),
  CREATETIME DATE,
  STARTTIME DATE,
  ENDTIME DATE,
  FINALTIME DATE,
  REMINDTIME DATE,
  ACTIONURL VARCHAR2(256),
  PROCESSINSTID NUMBER,
  ACTIVITYINSTID NUMBER,
  STATESLIST VARCHAR2(256),
  TIMEOUTNUM NUMBER,
```

TIMEOUTNUMDESC VARCHAR2(512),  
 PROCESSINSTNAME VARCHAR2(256),  
 ACTIVITYINSTNAME VARCHAR2(256),  
 PROCESSDEFID NUMBER,  
 PROCESSDEFNAME VARCHAR2(256),  
 PROCESSCHNAME VARCHAR2(256),  
 ACTIVITYDEFID VARCHAR2(64),  
 ASSISTANT VARCHAR2(64),  
 BIZSTATE NUMBER(\*,0),  
 ALLOWAGENT VARCHAR2(10),  
 ROOTPROCINSTID NUMBER,  
 ACTIONMASK VARCHAR2(64),  
 PARTINAME VARCHAR2(64),  
 ASSISTANTNAME VARCHAR2(64),  
 URLTYPE VARCHAR2(20),  
 DEALRESULT VARCHAR2(512),  
 DEALOPINION VARCHAR2(256),  
 EXTEND1 VARCHAR2(64),  
 EXTEND2 VARCHAR2(64),  
 EXTEND3 VARCHAR2(64),  
 EXTEND4 VARCHAR2(64),  
 EXTEND5 NUMBER,  
 CATALOGUUID VARCHAR2(32),  
 CATALOGNAME VARCHAR2(64),  
 TENANT\_ID VARCHAR2(128),  
 PRESSNUM NUMBER(32,0),  
 CONSTRAINT **PK\_WFWIPARTICIPANT1** PRIMARY KEY (WIPARTICID)  
 USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  
 STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  
 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1

```

BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE TS_INDX ENABLE
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE TS_WFP

```

```

partition by range (CREATETIME) INTERVAL (NUMTODSINTERVAL(1,'day')) store in
(TS_WFP)

```

```

(
partition P20180910 values less than (to_date('20180910','yyyymmdd')) tablespace
TS_WFP
);

```

上述黄色部分都是变化部分

#### 2.2.4 在中间表上创建索引

使用在线重定义进行转换时，原表上索引不会被自动转换过去，所以需要在中间表上创建索引，创建的索引是通过原表索引来进行创建的。

- 1) 先查询原表上的索引

```

Select index_name from dba_indexes where owner='CMSPRD' and table_name='
WFWIPARTICIPANT';

```

- 2) 收集原表索引信息

```

Set long 20000

```

```

Select dbms_metadata.get_ddl('INDEX',' IDX_WFWIPART_WIID',' CMSPRD') from dual;

```

。 。 。

- 3) 将上述索引重新命名，并在中间表上建立对应索引

```

CREATE INDEX CMSPRD.IDX_WFWIPART_WIID1 ON
CMSPRD.WFWIPARTICIPANT_TEMP01 (WORKITEMID) TABLESPACE TS_INDX;

```

```

CREATE UNIQUE INDEX CMSPRD.PK_WFWIPARTICIPANT1 ON
CMSPRD.WFWIPARTICIPANT_TEMP01 (WIPARTICID) TABLESPACE TS_INDX;

```

```

CREATE INDEX CMSPRD.WF_IDX_PART_CURRENTSTATE1 ON
CMSPRD.WFWIPARTICIPANT_TEMP01 (CURRENTSTATE) TABLESPACE TS_INDX;

```

```
CREATE INDEX CMSPRD.WF_IDX_PART_GLOBALID1 ON  
CMSPRD.WFWIPARTICIPANT_TEMP01 (GLOBALID) TABLESPACE TS_INDX;
```

```
CREATE INDEX CMSPRD.WIP_TENANT1 ON CMSPRD.WFWIPARTICIPANT_TEMP01  
(TENANT_ID) TABLESPACE TS_INDX;
```

### 2.2.5 开始重定义

```
call dbms_redefinition.start_redef_table('CMSPRD', 'WFWIPARTICIPANT',  
'WFWIPARTICIPANT_TEMP01');
```

### 2.2.6 同步数据

```
call dbms_redefinition.sync_interim_table('CMSPRD','WFWIPARTICIPANT',  
'WFWIPARTICIPANT_TEMP01');
```

### 2.2.7 完成重定义

```
call dbms_redefinition.finish_redef_table('CMSPRD','WFWIPARTICIPANT',  
'WFWIPARTICIPANT_TEMP01');
```

### 2.2.8 验证表定义是否被转为分区表

Set long 20000

```
Select dbms_metadata.get_ddl('TABLE',' WFWIPARTICIPANT',' CMSPRD') from dual;
```

### 2.2.9 确认是否存在状态为 **unusable** 的

```
SELECT OWNER,index_name,tablespace_name FROM DBA_INDEXES where owne=  
'CMSPRD' and status='UNUSABLE' order by 1,2;
```

--如果有，按照下面方法，生成批量重编译的句子，并重新编译。

```
select 'alter index '||owner||'.'||index_name||' rebuild tablespace TS_INDX;' RBINDX  
from dba_indexes where owner= 'CMSPRD' and status='UNUSABLE';
```

### 2.2.10 删除在线重定义生成的临时表

建议适当保留一段时间后，再来清除这个中间表

```
Drop table CMSPRD.WFWIPARTICIPANT_TEMP01 purge;
```

## 2.3 主键索引改为哈希索引

在压测过程中出现过主键索引产生索引争用的等待事件，为解决此类问题，把每个用户的主要一个表的主键索引改为哈希索引。所有用户的主键索引名字都相同，下面拿一个用户的改造方法举例：CMFPRD.PK\_WFPROCESSINST。

### 2.3.1 先确认要改造的主键索引和约束的名字

```
col constraint_name for a30
```

col constraint\_type for a10

col index\_name for a20

set line 120

col table\_name for a20

col partition\_name for a20

```
Select owner,constraint_name,constraint_type,index_name,table_name from
dba_constraints where index_name='PK_WFPROCESSINST' and owner='CMFPRD' order by 1;
```

OWNER TABLE_NAME	CONSTRAINT_NAME	CONSTRAINT	INDEX_NAME
-----			
AAMPRD WFPROCESSINST	PK_WFPROCESSINST	P	PK_WFPROCESSINST
CMFPRD WFPROCESSINST	PK_WFPROCESSINST	P	PK_WFPROCESSINST
CMSPRD WFPROCESSINST	PK_WFPROCESSINST	P	PK_WFPROCESSINST
ERPPRD WFPROCESSINST	PK_WFPROCESSINST	P	PK_WFPROCESSINST
ITPRD WFPROCESSINST	PK_WFPROCESSINST	P	PK_WFPROCESSINST
MDMPRD WFPROCESSINST	PK_WFPROCESSINST	P	PK_WFPROCESSINST
SOAPRD WFPROCESSINST	PK_WFPROCESSINST	P	PK_WFPROCESSINST
TAXPRD WFPROCESSINST	PK_WFPROCESSINST	P	PK_WFPROCESSINST
WFPPRD WFPROCESSINST	PK_WFPROCESSINST	P	PK_WFPROCESSINST

如上，约束和索引的名字都是相同的：PK\_WFPROCESSINST。

### 2.3.2 删除已有主键约束

```
ALTER TABLE CMFPRD.WFPROCESSINST DROP CONSTRAINT PK_WFPROCESSINST;
```

### 2.3.3 创建唯一索引

```
CREATE UNIQUE INDEX CMFPRD.PK_WFPROCESSINST ON CMFPRD.WFPROCESSINST  
(PROCESSINSTID) tablespace ts_indx GLOBAL PARTITION BY HASH (PROCESSINSTID)  
PARTITIONS 36;
```

注意创建的索引名字和原来名字要相同。

### 2.3.4 增加主键约束

```
alter table CMFPRD.WFPROCESSINST add constraint PK_WFPROCESSINST primary key  
(PROCESSINSTID) using index CMFPRD.PK_WFPROCESSINST;
```

### 2.3.5 调整各个子分区的存储属性

```
select 'alter index CMFPRD.PK_WFPROCESSINST rebuild partition ' || partition_name ||'  
PARAMETERS ("pctfree=35") tablespace TS_INDx;' cmd_list from dba_ind_partitions where  
index_name='PK_WFPROCESSINST' and index_owner='CMFPRD';
```

将上述结果进行执行。

### 2.3.6 --各分区状态确认

```
select index_owner,partition_name,status from dba_ind_partitions where  
index_name='PK_WFPROCESSINST' and index_owner in
```

```
('RBS',  
'ERPPRD',  
'ITPRD',  
'CMSPRD',  
'SOAPRD',  
'MDMPRD',  
'AAMP RD',  
'TAXPRD',  
'CMFPRD',  
'WFPPRD') order by index_owner,partition_name;
```

如果有异常状态的，需要重新进行各分区的编译，可按照下面脚本生成批量脚本：

---

```
select 'alter index ' || index_owner || '.' || 'PK_WFPROCESSINST rebuild partition '  
|| partition_name || ' PARAMETERS ("pctfree=35") tablespace TS_INDx;' cmd_list  
  
from dba_ind_partitions where index_name='PK_WFPROCESSINST' and  
  
index_owner='CMSPRD';
```

## 2.4 2节点数据未完成工作

用户：WFPBPPRD 未改成哈希索引，可按照下面3步进行操作：

Conn /as sysdba

Sql> 环境下执行下面3个语句，如果语句报异常，说明对象被占用，建议暂时停止业务后再尝试执行。

- 1) ALTER TABLE WFPBPPRD.WFPROCESSINST DROP CONSTRAINT PK\_WFPROCESSINST;
- 2) CREATE UNIQUE INDEX WFPBPPRD.PK\_WFPROCESSINST ON WFPBPPRD.WFPROCESSINST (PROCESSINSTID) tablespace ts\_indx GLOBAL PARTITION BY HASH (PROCESSINSTID) PARTITIONS 36;
- 3) alter table WFPBPPRD.WFPROCESSINST add constraint PK\_WFPROCESSINST primary key (PROCESSINSTID) using index WFPBPPRD.PK\_WFPROCESSINST;

---

## 3. 参考命令

重建普通索引，重建到TS\_INDX表空间

```
alter index owner.index_name rebuild tablespace TS_INDX;
```

重建LOB索引,移动到TS\_INDX表空间

```
alter table OWNER.TABLE_name move lob(column_name) store as (tablespace TS_INDX);
```