

*Online Database Block Size Rebuilds
in Oracle9i*

Kevin Loney, TUSC

<http://www.tusc.com>

800-755-TUSC

TUSC Service Offerings

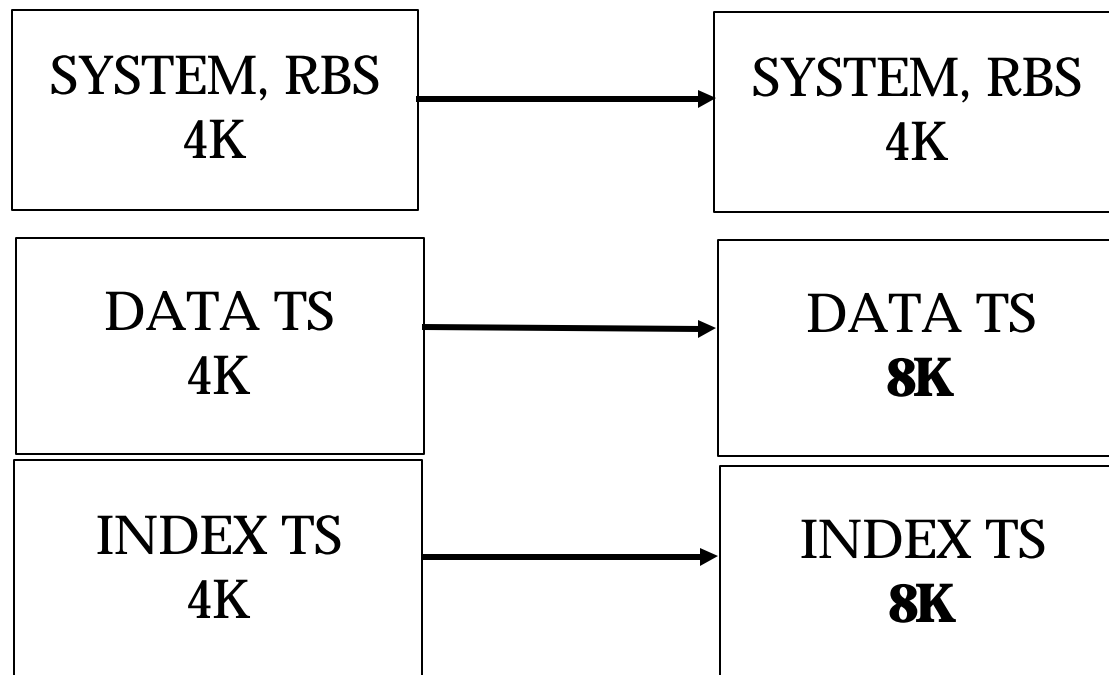
- Database Services
- Remote Database Administration
- Expert Solutions Practice
- Oracle Applications
 - CRM/ERM Implementations/Upgrades
- Training and Mentoring

Remote Database Administration

- Remote*DBA is a proactive database monitoring service designed to provide customers a cost-effective approach for maximizing the availability and performance of their Oracle databases.
- Remote*DBA provides the expertise and monitoring software that allows TUSC expert DBAs to remotely diagnose and manage Oracle databases regardless of their location.

Primary goal for today

- Increase the database block size in a live database while it is being accessed



Secondary goals

- Demonstrate online reorgs, dynamic memory management, and Oracle-managed files
- Show how to monitor the contents of the memory areas

Tertiary goal

- Avoid a database crash during the presentation.

```
if crash:= 'YES'  
  blame:= 'NT';  
endif;
```

Why increase the block size?

- Significant advantage for range query performance
- Significant advantage for bulk insert performance
 - Reduces the index B*levels for the same number of rows
 - For OLTP, can manage potential conflicts via FREELISTS, INITRANS, MAXTRANS settings

Oracle9i Features

- Features to exercise:
 - DB block size at the tablespace level
 - Separate buffer caches per block size
 - Dynamically changing buffer cache sizes
 - Oracle-managed files
 - Online object reorganizations
- Not exercising: Quiesce

Quiesce

- When the database is quiesced, no transactions by users other than SYS or SYSTEM are permitted
- All actions appear to hang, including logins

```
alter system quiesce restricted;
```

Is it Quiesced?

```
select Active_State  
from V$INSTANCE;
```

- values: NORMAL, QUIESCED,
QUIESCING

```
alter system unquiesce;
```

Why not use quiesce?

- Requires that all actions be executed by SYS or SYSTEM
 - Impacts ownership of indexes
 - Impacts creation of foreign keys and links
 - Impacts availability of the database

SGA, 9i style

```
select Name, Value  
from V$PARAMETER  
where Name like '%cache_size';
```

- **DB_BLOCK_BUFFERS** is deprecated.

Sample output

NAME	VALUE
db_keep_cache_size	0
db_recycle_cache_size	0
db_2k_cache_size	0
db_4k_cache_size	0
db_8k_cache_size	0
db_16k_cache_size	0
db_32k_cache_size	0
db_cache_size	20971520

SGA_MAX_SIZE

- The *SGA_MAX_SIZE* parameter sets the maximum combined size for the SGA areas (including shared pool)
- *DB_nK_CACHE_SIZE* should be 0 for the database's created block size
- Cache sizes may be altered - increments are called granules

Granule sizes

- If `SGA_MAX_SIZE` < 128 M, granule size = 4 MB.
- Otherwise, granule size = 16 MB.
- Other values are dynamically rounded up to the next highest granule size.

- For this example, `DB_BLOCK_SIZE` = 4096, `SGA_MAX_SIZE`=105672596

Minimum sizes for an NT install

- `shared_pool_size = 60m`
- `db_cache_size = 20m`
- `java_pool_size = 8m`
 - If `java_pool_size = 4m`, install will complete but all subsequent DDL operations will return shared pool errors (fixed in 9i Release 2)
 - Changing `java_pool_size` or `sga_max_size` requires a shutdown/startup

Using OMF to simplify example

- Oracle-managed files are automatically created and deleted when a tablespace is created or dropped.
- OMFs can also include control files and redo log files.
- Can create an entire database with:
`create database MYDB;`

File locations

- All datafiles:

DB_CREATE_FILE_DEST='/u01/oracle/CC1'

- 5 redo log and control file directories:

DB_CREATE_ONLINE_LOG_DEST_1='/u02/oracle/CC1'

DB_CREATE_ONLINE_LOG_DEST_2='/u03/oracle/CC1'

...

Default sizes

- **SYSTEM ts: 100 MB, autoextensible**
- **Redo logs: 100 MB each**
- **Undo tablespace: 10 MB autoextensible**
- **Default temp ts: 100 MB autoextensible**

- **User tablespaces: 100 MB autoextensible**

File names

- File names will include a system-generated unique identifier:

`/u01/oracle/CC1/users_dest_erj42201.dbf`

- OMF files can be renamed using standard methods (alter database rename file etc) following their creation.

Specifying block size for a tablespace

- Override the DB_BLOCK_SIZE at the tablespace level:

```
create tablespace NEW_TS  
blocksize 8K;
```

Example, step 1 - create a cache

- Increase `SGA_MAX_SIZE` if necessary
 - ideally keep at least 1 granule of free cache space available, or you will need to shutdown/startup
- Create the cache for the new block size

```
alter system set db_8k_cache_size=4M;
```

Example, Step 2 - create tablespace

```
alter system set  
DB_CREATE_FILE_DEST='E:\Oracle\oradataomf\';
```

- **In USERS (4K block size): SCOTT.EMP**

```
create tablespace USERS_DEST  
datafile size 5M blocksize 8k;
```

```
alter user SCOTT quota 5M  
on USERS_DEST;
```

Verify the file creation & block size

- Check the directory
 - Rename the file if you wish

```
select Block_Size
from DBA_TABLESPACES
where Tablespace_Name='USERS_DEST' ;
```

```
BLOCK_SIZE
-----
          8192
```


Note on cache/tablespace

- While an 8K tablespace exists, the `db_8k_cache_size` parameter cannot be set back to 0.

Checklist

- Cache altered
- Tablespace with higher block size created
- OMF created

- Next: Reorg the table
 - Restrictions apply.

Restrictions include

You cannot perform online reorganizations for tables with no primary keys, for tables that have materialized views and materialized view logs defined on them, for materialized view container tables, for advanced queueing tables, SYS/SYSTEM tables, tables with LONGs, or for IOT overflow tables.

- See the Oracle9i Admin Guide for full list!

Step 3 - Redefinition checklist

- 3.1. Verify table can be rebuilt online
- 3.2. Create interim table
 - indexes, grants, constraints, triggers
- 3.3. Start the redefinition
- 3.4. Optionally sync the source/destination tables or abort the process
- 3.5. Finish the redefinition

3.1 Can the table be rebuilt online?

execute

```
DBMS_REDEFINITION.CAN_REDEF_TABLE  
( 'SCOTT' , 'EMP' ) ;
```

- Errors will give the exceptions encountered.

Errors may confuse users

*

```
ERROR at line 1:
```

```
ORA-00942: table or view does not exist
```

```
ORA-06512: at "SYS.DBMS_REDEFINITION"  
          line 8
```

```
ORA-06512: at "SYS.DBMS_REDEFINITION"  
          line 236
```

```
ORA-06512: at line 1
```

- What is the problem?

3.2 Create the interim table

```
create table EMP_DEST
(EMPNO      NUMBER(4) PRIMARY KEY,
 ENAME      VARCHAR2(10),
 JOB        VARCHAR2(9),
 MGR        NUMBER(4),
 HIREDATE   DATE,
 SAL        NUMBER(7,2),
 COMM       NUMBER(7,2),
 DEPTNO     NUMBER(2)) . . .
```

Partition during the move

```
... partition by range (DeptNo)
(partition PART1
  values less than ('30'),
partition PART2
  values less than (MAXVALUE))
tablespace USERS_DEST ←
storage (freelists 5); ←
```


What EMP_DEST is missing

- Grants
- Triggers
- Indexes
- Foreign keys
 - Create as disabled until process is completed
- Can add all of these prior to starting the redefinition process

At the end of the process:

- The grants, indexes, triggers, and constraints on the EMP_DEST table will replace those from the original table.
- Can place indexes in their own separate 8K block size tablespace

3.3 Start the redefinition

- Specify column mapping as the fourth parameter if the columns in the interim table are different from the source table.

```
execute  
DBMS_REDEFINITION.START_REDEF_TABLE -  
  ( 'SCOTT' , 'EMP' , 'EMP_DEST' ) ;
```

Quick check

```
select COUNT(*)  
  from EMP_DEST;
```

```
  COUNT(*)  
-----  
          14
```

While the rebuild is occurring,

- Transactions may be executed against the source table at any time.

```
insert into EMP values  
  (1111, 'FIRST', 'CEO', NULL,  
   '01-JAN-01', 1, 1, 10);  
commit;
```

3.4 - OPTIONAL - Abort

- To abort the redefinition process:

execute

```
DBMS_REDEFINITION.ABORT_REDEF_TABLE -  
  ( ' SCOTT ' , ' EMP ' , ' EMP_DEST ' ) ;
```

- May want to truncate EMP_DEST
- Can later restart from Step 3.3

3.4 *OPTIONAL - Sync*

- Sync EMP and EMP_DEST to make the data changes in the final step complete faster
 - Offline instantiation of the committed data

execute

```
DBMS_REDEFINITION.SYNC_INTERIM_TABLE -  
( 'SCOTT' , 'EMP' , 'EMP_DEST' ) ;
```

Post-SYNC

```
select COUNT(*)  
  from EMP_DEST;
```

```
COUNT(*)
```

```
-----
```

```
15
```


3.5 Finish the redefinition

execute

```
DBMS_REDEFINITION.FINISH_REDEF_TABLE -  
  ( 'SCOTT' , 'EMP' , 'EMP_DEST' ) ;
```

- **Minimize the number of transactions between START and FINISH**

4. *Verify the redefinition*

```
select Table_Name, Tablespace_Name,  
       High_Value  
from DBA_TAB_PARTITIONS  
where Owner = 'SCOTT';
```

TABLE_NAME	TABLESPACE_NAME	HIGH_VALUE
EMP	USERS_DEST	MAXVALUE
EMP	USERS_DEST	'30'

The new EMP

- EMP is now in USERS_DEST
 - 8k block size
- EMP blocks will now use the db_8k_cache_size cache
- EMP is now partitioned

Dealing with the reorganized table

- Check that foreign keys are enabled
- Check grants
- Create non-primary key indexes
- Rebuild materialized views

5. Dealing with the interim table

- EMP_DEST is still out there!

```
select COUNT(*)  
  from EMP_DEST;
```

```
  COUNT(*)  
-----  
          15
```

Interim table, post-redefinition

- EMP_DEST is no longer partitioned!
 - As SCOTT:

```
select Distinct Table_Name  
from USER_TAB_PARTITIONS;
```

```
TABLE_NAME
```

```
-----
```

```
EMP
```

After verification:

- **Truncate or drop EMP_DEST**
 - It switched names with the old EMP table
 - Located in the old EMP table's tablespace

Now what?

- Expand to full tablespace-level reorganizations
 - `START_REDEF_TABLE` on multiple tables in parallel
 - Group application tables by tablespace
 - Allocate space for source and interim tables to be online concurrently

6. Additional options - partitions

- Move partitioned tables via existing ALTER TABLE options,
no DBMS_REDEFINITION calls necessary

For indexes

- As of Oracle 8i, you can also use **ALTER INDEX REBUILD ONLINE**
 - No support for bitmap indexes
 - Known index corruption bugs for this process in 8i.
 - Until 9i, no support for functional indexes, reverse key indexes, compressed key indexes, etc.

7. Monitoring usage of the caches

- Identify object usage by block size cache

What's in the buffer cache?

- In single-block size databases or the default cache, join X\$BH to DBA_OBJECTS

```
select  Object_Name,
        Object_Type ,
        COUNT(*) Num_Buff
from X$BH A, SYS.DBA_OBJECTS B
where A.Obj = B.Object_ID
and Owner not in ('SYS', 'SYSTEM')
group by Object_Name, Object_Type;
```

What's in multi block size caches?

- X\$BH contains the buffer cache data for all block sizes. The BLSIZ column shows the block size (4096, 8192, etc).
- For non-default block sizes, the Obj values in X\$BH do not match Object_Ids in DBA_OBJECTS.

X\$BH

```
select distinct Obj from X$BH where BlSiz=8192;
```

```
          OBJ  
-----  
          0  
        31540  
        31541
```

```
select MAX(Object_ID) from DBA_OBJECTS;
```

```
MAX(OBJECT_ID)  
-----  
        31519
```

EMP

- EMP now has 3 object_Ids:
 - 31516 table
 - 31517 and 31518 table partitions

Use extent location instead

- **Slower than joining X\$BH to DBA_OBJECTS**

```
select Segment_Name, Segment_Type,  
       COUNT(*) Num_Buff  
from X$BH a, SYS.DBA_EXTENTS B  
where a.Dbarfil = b.File_id  
and a.Dbablk >= b.Block_id  
and a.Dbablk <= (b.Block_id + b.Blocks)  
and Owner not in ('SYS', 'SYSTEM')  
and a.Blsiz=8192  
group by Segment_Name, Segment_Type;
```


What's in the 8K cache

SEGMENT_NAM	SEGMENT_TYPE	NUM_BUFF
EMP	TABLE PARTITION	7
SYS_C002451	INDEX	1

To see all, add Blsiz and Owner

```
select Blsiz, Owner, Segment_Name,  
       Segment_Type, COUNT(*) Num_Buff  
from X$BH a, SYS.DBA_EXTENTS B  
where a.Dbarfil = b.File_id  
and a.Dbablk >= b.Block_id  
and a.Dbablk <= (b.Block_id + b.Blocks)  
and Owner not in ('SYS', 'SYSTEM')  
group by Blsiz, Owner, Segment_Name,  
Segment_Type;
```

- May wish to also exclude owners WKSYS, SH, RMAN, etc

Looking back

- **DB_nK_BLOCK_SIZE** - set up new caches and tablespaces online
- Used Oracle-Managed Files
- Reorganized tables into higher block sizes online
- Partitioned non-partitioned tables online
- Verified cache usage

In the interest of full disclosure...

- Oracle problems encountered:
 - minimum cache sizes during installation
 - java_pool_size bug
- NT bugs encountered:
 - Required over-allocation of virtual memory
 - Repeated OleMainThreadWndName services problems, resolved by disabling “Allow service to interact with desktop” from Services panel

Questions?



loneyk@tusc.com
<http://www.tusc.com>
1-800-755-TUSC

www.tusc.com