



NMHC RX

Pharmacy Benefits Manager



NMHC_{RX}

Pharmacy Benefits Manager

WELCOME TO



***GO LOCAL ! - ALL ABOUT
LOCALLY MANAGED
TABLESPACES***

A Seminar for LISIG

BATMAN & ROBIN

By **Raj Sakthi**

JUNE 1997



Audience

- Developers
- DBAs

Scope

- Knowledge Assumed
- What this is not..!



Overview



- **LMT – A 30,000 Feet View**
- **So what's wrong with DMT ?**
- **What LMT is not !**
- **LMT – Getting Personal**
- **Tradeoffs**
- **Conclusion**

LMT – A 30,000 Feet View

Space management 101

- Space management in a tablespace can be reduced to
 - Tracking which Extents belong to which Segments.
 - Tracking which extents are free in a Tablespace and which are used.
- Historically, both of these have been achieved though some data dictionary tables like SEG\$, in case of former, and FET\$ and UET\$, in case of latter.
- The former has not changed in methodology yet, the latter has – Enter LMT.

Meet LMT – Better Than meeting Joe Black...!(Contd)

- In a **Locally Managed Tablespace**, the space management is carried out using 64K bitmaps in the header of each data files.
- Each Bit in bitmap represents group of Blocks in file.
- If the bit is set then the group of block it corresponds to is in use , if it is clear then the group is free.
- Inside LMT the extents could be allocated in 2 different manners
 - AUTOALLOCATE
 - UNIFORM

Meet LMT (Contd)

- Creation of LMT is simple enough.
 - Management type is specified at the time of creation
 - In 8i Default is DMT
 - In 9i Default is LMT (with compatible set to 9.0.0)

- **CREATE TABLESPACE LMT_DEMO**
DATAFILE '< >' **SIZE** < >
EXTENT MANAGEMENT LOCAL
UNIFORM SIZE < > [**AUTOALLOCATE**]

Meet LMT (Contd)

- **AUTOALLOCATE.**
 - Option slices up the data file into 64K slices, but extents are allocated based on pattern of usage and gaps available in the file, always from the list – 64K, 1MB, 8MB, 64MB OR 256MB.
 - Can not use this option for Rollback segment Tablespace and Temp.(This restriction is lifted in 9.2.0)
- **UNIFORM SIZE.**
 - Data file is sliced into equal sized extents with the exception of 64K header.

So What's wrong with DMT ?

- Under **D**ictionary **M**anaged **T**ablespace, whenever new extent needs to be allocated, system
 - Checks if free space is available by querying FET\$.
 - Deletes or Modifies a row from FET\$ and insert a row into UET\$.
 - May insert or modify a row in SEG\$ and TSQ\$.
- All this 'recursive SQL' generates redo.
- All space management is protected by single Space Transaction ('ST') enqueue. This might lead to occasional concurrency issue.

So What's wrong with DMT ?

(Contd)

- Other complaints about DMT
 - Fragmentation
 - Creating or Dropping objects with large number of extents can take a long time.
 - Bad Default values.
 - Objects can inherit space parameters from Tablespace default settings, which may not be suitable for that object.
 - ‘Folk wisdom’ passing on bad advises like containing all the data in ONE extent and setting PCTINCREASE to 1 to coalesce space.

Bursting some Bubbles - What LMT is not !

- It is advisable to move to LMT – but not for the reasons you might think..!
- Fragmentation – Could be avoided by
 - Using equal sized extents – same INITIAL and NEXT.
 - Set PCTINCREASE to zero.
 - Additional insurance – equal extent size can be forced at the tablespace level by setting oh-so-rarely-used parameter ‘minimum extent’ equal to INITIAL and NEXT.
- ST Lock issue – How many times in a day you would be doing large number of allocation and deallocations to worry about ST enqueue ?

Bursting some Bubbles - What LMT is not !(Contd)

- RECURSIVE SQL - Although FET\$ and UET\$ is no longer used to track space information, recursive sql is still needed to access and modify SEG\$ and TSQ\$.
- Bitmap Operations are not blazing fast as touted and were comparable in speed with the DML based storage manipulation of DMT.

LMT – Getting Personal

- Alright you are saying LMT is not a silver bullet that kills the space Vampire. Well...what exactly LMT is good for again..?
 - Under **Uniform size** , all the extents are same size. Since only major Space admin action can bend/break this pattern, any chance of ‘whimsical’ space allocation is avoided.
 - No redo for bitmap modifications.
 - No more ST locks needed to protect space management operations of LMT. The TT Lock, one per tablespace, is used instead.

LMT – Getting Personal (Contd)



- Objects with large number of extents can be dropped or created with out any significant performance degradation.
- Free space management is very efficient by not needing to coalesce adjacent free extents i.e. No more honey comb fragmentation.
- Standby Databases opened in a read only mode can have temporary tablespaces for sorting, group by etc. for reporting.

LMT – Getting Personal (Contd)

- From 9i onwards ORACLE Corp seems to be pushing LMT. Under 9.2.0 the Database Creation assistance, by default ,creates database with SYSTEM table space as LMT.
- There is nothing wrong with that except for the fact that if the system table space is created as LMT, then database *can not* have a DMT unless it is a ‘plugged in’ table space and that too only when it is read only.

LMT – Getting Personal (Contd)

- ‘Sweat the small stuff’

- The size declaration of LMT data file is worth some mentioning as it is source of common mistake of most. If you don't add 64K to the desired size you will end up with one extent short.
 - `CREATE TABLESPACE LMT_DEMO
DATAFILE '/dev_u01/oradata/zeta/lmt_demo_1.dbf' SIZE 2048K
EXTENT MANAGEMENT LOCAL
UNIFORM SIZE 1024K ;`
- Lets select the free space available in this Table space.

LMT – Getting Personal (Contd)

- ‘Sweat the small stuff’

Oracle SQL*Plus

File Edit Search Options Help

SYS@ZETA@RSAKTHI>

SYS@ZETA@RSAKTHI>

SYS@ZETA@RSAKTHI>

SYS@ZETA@RSAKTHI>

SYS@ZETA@RSAKTHI>

```
SYS@ZETA@RSAKTHI>select tablespace_name,file_id,block_id,bytes/1024 "Bytes in K",blocks
 2  from dba_free_space
 3  where tablespace_name like 'RAJ%'
 4  /
```

TABLESPACE_NAME	FILE_ID	BLOCK_ID	Bytes in K	BLOCKS
RAJ_LMT_TEST	8	9	1024	128

SYS@ZETA@RSAKTHI>

SYS@ZETA@RSAKTHI>

SYS@ZETA@RSAKTHI>

LMT – Getting Personal (Contd)

- ‘Sweat the small stuff’

- In the above case we have ended up with only one extent of size 1024K. This is because system tries to carve data file into equal sized extents of 1024K plus 64K header. So when first 64K is taken for header and rest is found to be smaller than the required 1024K, that is left unused .
- I got an idea...! instead of feeling warm and ‘fussy’, why not simplify the whole thing and just add 1 MB to the desired size.

LMT – Getting Personal (Contd)

- ‘Sweat the small stuff’

- The following are the minimum limits
 - On LMT with *UNIFORM SIZE*, the extents must be at least 2 blocks in size. If not system silently makes it so.
 - Smallest table space must be equal to (1 extent + 2 blocks + 1 block for bitmap) in size.
 - Yes...I said bitmap of size 1 block...!! When you are creating small table space ORACLE gives you only one block bitmap.

TABLESPACE_NAME	FILE_ID	BLOCK_ID	BYTES	BLOCKS
RAJ_LMT_TEST	8	4	16384	2

- Just for giggles, in a regular LMT ,the max number of extents a data file can have before creating additional bitmap is 380,000 for a database with 8K block size.

LMT – Getting Personal (Contd)

- ‘Sweat the small stuff’

Connected to:
Oracle8i Enterprise Edition Release 8.1.6.3.0 - Production
With the Partitioning option
JServer Release 8.1.6.3.0 - Production

```
SYS@ZETA@RSAKTHI>CREATE TABLESPACE RAJ_LMT_TEST
 2     DATAFILE 'C:\ORA816\ORA81\ORADATA\ZETA\RAJ_LMT_TEST_01.DBF' SIZE 33K reuse
 3     EXTENT MANAGEMENT LOCAL uniform size 16K
 4     NOLOGGING
 5     ONLINE
 6 /
```

Tablespace created.

```
SYS@ZETA@RSAKTHI>select tablespace_name,file_id,block_id,bytes/1024 "Bytes in K",blocks
 2 where tablespace_name like 'RAJ%'
 3 ;
```

TABLESPACE_NAME	FILE_ID	BLOCK_ID	Bytes in K	BLOCKS
RAJ_LMT_TEST	8	4	16	2

```
SYS@ZETA@RSAKTHI>|
```

LMT – Getting Personal (Contd)

- ‘Sweat the small stuff’

- Object creation with INITIAL specified is possible only on LMT with *autoallocate*. System acts like a good sport and accepts your initial and silently rounds it off to multiples of 64K.
- Under autoallocate the extents start off as 64K and when they reach 1 MB (16*64K) the next extent size changes to 1MB.
- When Table increases in size to 64MB the next extent becomes 8MB in size and so on.

LMT – Getting Personal (Contd)

- ‘Sweat the small stuff’

```
Oracle SQL*Plus
File Edit Search Options Help
SYS@ZETA@RSAKTHI>
SYS@ZETA@RSAKTHI>
SYS@ZETA@RSAKTHI>SELECT segment_name,tablespace_name,extent_id,bytes/1024 "Bytes in K"
 2 FROM dba_extents
 3 where segment_name ='TEST'
 4 ;
```

SEGMENT_NAME	TABLESPACE_NAME	EXTENT_ID	Bytes in K
TEST	RAJ_LMT_TEST	0	64
TEST	RAJ_LMT_TEST	1	64
TEST	RAJ_LMT_TEST	2	64
TEST	RAJ_LMT_TEST	3	64
TEST	RAJ_LMT_TEST	4	64
TEST	RAJ_LMT_TEST	5	64
TEST	RAJ_LMT_TEST	6	64
TEST	RAJ_LMT_TEST	7	64
TEST	RAJ_LMT_TEST	8	64
TEST	RAJ_LMT_TEST	9	64
TEST	RAJ_LMT_TEST	10	64
TEST	RAJ_LMT_TEST	11	64
TEST	RAJ_LMT_TEST	12	64
TEST	RAJ_LMT_TEST	13	64
TEST	RAJ_LMT_TEST	14	64
TEST	RAJ_LMT_TEST	15	64
TEST	RAJ_LMT_TEST	16	1024

17 rows selected.

LMT – Getting Personal (Contd)

- ‘Sweat the small stuff’

- *When creating table on a LMT with *Autoallocate*, if you specify MINEXTENT greater than 1, table creation will **Not True**
 - Fail giving misleading error.
 - ORA-01658: unable to create INITIAL extent
 - Workaround : create the table with only one extent on the tablespace in autoallocation method, and then force extent allocation.

LMT – Getting Personal (Contd)

- ‘Sweat the small stuff’

Oracle SQL*Plus

File Edit Search Options Help

SYS@ZETA@RSAKTHI>

SYS@ZETA@RSAKTHI>

SYS@ZETA@RSAKTHI>CREATE TABLESPACE RAJ_LMT_TEST

2 DATAFILE 'C:\ORA816\ORA81\ORADATA\ZETA\RAJ_LMT_TEST_01.DBF' SIZE 4M reuse

3 EXTENT MANAGEMENT LOCAL autoallocate

4 NOLOGGING

5 ONLINE

6 /

Tablespace created.

SYS@ZETA@RSAKTHI>

SYS@ZETA@RSAKTHI>CREATE TABLE TEST(NAME VARCHAR2(10))

2 TABLESPACE RAJ_LMT_TEST

3 STORAGE(minextents 3)

4 ;

Table created.

SYS@ZETA@RSAKTHI>

SYS@ZETA@RSAKTHI>SELECT segment_name,tablespace_name,extent_id,bytes/1024 "Bytes in K"

2 FROM dba_extents

3 where segment_name = 'TEST';

SEGMENT_NAME	TABLESPACE_NAME	EXTENT_ID	Bytes in K
TEST	RAJ_LMT_TEST	0	1024
TEST	RAJ_LMT_TEST	1	1024
TEST	RAJ_LMT_TEST	2	1024

LMT – Getting Personal (Contd)

- ‘Sweat the small stuff’

- The same on LMT with *Uniform Size* succeeds as follows.

LMT – Getting Personal (Contd)

- ‘Sweat the small stuff’

Oracle SQL*Plus

File Edit Search Options Help

```
SYS@ZETA@RSAKTHI>
```

```
SYS@ZETA@RSAKTHI>
```

```
SYS@ZETA@RSAKTHI>CREATE TABLESPACE RAJ_LMT_TEST
```

```
2     DATAFILE 'C:\ORA816\ORA81\ORADATA\ZETA\RAJ_LMT_TEST_01.DBF' SIZE 2048K reuse
```

```
3     EXTENT MANAGEMENT LOCAL uniform size 50K
```

```
4     NOLOGGING
```

```
5     ONLINE
```

```
6 /
```

Tablespace created.

```
SYS@ZETA@RSAKTHI>
```

```
SYS@ZETA@RSAKTHI>CREATE TABLE TEST
```

```
2 (
```

```
3   NAME VARCHAR2(10)
```

```
4 )
```

```
5 TABLESPACE RAJ_LMT_TEST
```

```
6 STORAGE(minextents 3)
```

```
7 ;
```

Table created.

LMT – Getting Personal (Contd)

- ‘Sweat the small stuff’

Oracle SQL*Plus

File Edit Search Options Help

SYS@ZETA@RSAKTHI>

SYS@ZETA@RSAKTHI>

```
SYS@ZETA@RSAKTHI>SELECT table_name,tablespace_name,initial_extent/1024,next_extent/1024
 2 FROM DBA_TABLES
 3 WHERE TABLE_NAME = 'TEST'
 4 ;
```

TABLE_NAME	TABLESPACE_NAME	INITIAL_EXTENT/1024	NEXT_EXTENT/1024
TEST	RAJ_LMT_TEST	168	56

SYS@ZETA@RSAKTHI>

```
SYS@ZETA@RSAKTHI>SELECT segment_name,tablespace_name,extent_id,bytes/1024 "Bytes in K"
 2 FROM dba_extents
 3 where segment_name ='TEST'
 4 /
```

SEGMENT_NAME	TABLESPACE_NAME	EXTENT_ID	Bytes in K
TEST	RAJ_LMT_TEST	0	56
TEST	RAJ_LMT_TEST	1	56
TEST	RAJ_LMT_TEST	2	56

SYS@ZETA@RSAKTHI>

LMT – Getting Personal (Contd)

- ‘Sweat the small stuff’

- If you try to modify NEXT storage parameter, which is not normally advised under LMT, it will error out with
 - *ORA-25150: Altering of extent parameters not permitted.*
- But if you must,
 - Migrate the locally-managed table space to a dictionary Management.
 - Migrate it back to Local.
 - This way the system records the extent allocation type to be *‘user’* and you can modify the next parameter.

LMT – Getting Personal (Contd)

- ‘Sweat the small stuff’

- ‘Converting’ from DMT to LMT – Not exactly a way to salvation..!
 - EXECUTE
SYS.DBMS_SPACE_ADMIN.TABLESPACE_MIGRATE_TO_LOCAL('ZETA');
- The extent allocation remains as ‘whimsically wild’ as it is in DMT meaning neither UNIFORM size nor AUTOALLOCATE is enforced.
- *Allocation_type* is set to ‘USER’ and allocation continues to be based on *next_extent*, with the consolation that system will round it off to *unit_size* determined during conversion.
- *Unit_size* is the ‘greatest common divisor’ for all the extents you have in the DMT.

LMT – Getting Personal (Contd)

- ‘Sweat the small stuff’

- Unlike ‘Pure bred’ LMTs , only one bitmap is created for all the data files , in the first data file or in the data file you specify during conversion. So table space being converted needs to have free extent of size $64K * (\text{No. of data files})$ in the first file or the file you are specifying.
- Temporary DMTs can not be converted to LMT

LMT – Getting Personal (Contd)

- ‘Sweat the small stuff’

- Just for the sake of completion, static views that gives information about LMT are
 - DBA_DATA_FILES
 - DBA_TABLESPACES

LMT - Trade offs

- Since the space tracking information is stored in bitmaps in the header of files and not in tables in data dictionary, space summary queries cause a whole lot of physical reads.
- Scanning Static view like DBA_EXTENTS, DBA_FREE_SPACE will be slower.
- Old dog – new trick problem..?

Parting Gift..?

- The following script summarises the space information for a hybrid environment – that is with both DMT and LMT

```
--This script reports the free space available in Tablespaces
--The neat thing about this is it reports the type of tablespace
--Right alogside of the name ..!!
--Raj Sakthi (raj_sakthi@yahoo.com)
--
set feedback off
set termout off
break on today
column today new_value dba_date
select to_char(sysdate, 'mm/dd/yy hh:mi AM') today
       from dual;
clear breaks
COL DB_N NEW_VALUE DB_NAME noprint
BREAK ON DB_NAME
select upper(value) db_n from v$parameter
       where name = 'db_name';
set termout on
set pagesize 999 linesize 150 verify off
set space 2
title left 'Date: ' format a18 dba_date skip 1-
center 'FREE SPACE SUMMARY IN DATABASE ' DB_NAME skip 1-
right 'Page: ' format 999 sql.pno skip 2
column dummy noprint
column pct_used format 999.99 heading "%|Used"
column pct_free format 999.99 heading "%|FREE"
column name format a28 heading "Tablespace Name"
column MBytes format 999,999.999 heading "MBytes"
```

Script (Contd)

```
column used format 999,999.99 heading "Used|MB"
column free format 999,999.99 heading "Free|MB"
column largest format 99,999.99 heading "Largest|MB"
break on report
compute sum of MBytes on report
compute sum of free on report
compute sum of used on report
select nvl(b.tablespace_name,nvl(a.tablespace_name,'UNKOWN'))||
-- decode(c.ext,'LOCAL','(LMT)','(DICT)') name,
decode(c.ext,'LOCAL','(LMT)') name,
    MBytes_alloc MBytes,
    MBytes_alloc-nvl(MBytes_free,0) used,
    nvl(MBytes_free,0) free,
    ((MBytes_alloc-nvl(MBytes_free,0))/MBytes_alloc)*100 pct_used,
    (Mbytes_free/MBytes_alloc)*100 pct_free,
    nvl(largest,0) largest
from ( select sum(bytes)/1024/1024 MBytes_free,
        max(bytes)/1024/1024 largest,
        tablespace_name
      from sys.dba_free_space
      group by tablespace_name ) a,
     ( select sum(bytes)/1024/1024 MBytes_alloc,
        tablespace_name
      from sys.dba_data_files
      group by tablespace_name ) b ,
(select extent_management ext ,tablespace_name from dba_tablespaces
 where max_extents is not null ) c
  where a.tablespace_name (+) = b.tablespace_name and
a.tablespace_name = c.tablespace_name (+)
order by 1

spool free_space_&&DB_NAME..doc
/
title off
SET HEADING OFF
select 'TEMP FILE LISTING OF DB &&DB_NAME ' from dual
where exists ( select 1 from dba_temp_files
having count(9) > 0 );
```

Script (Contd)

```
set heading on
col file_name form a50
break on tablespace_name skip 2 nodup
compute sum of "SIZE IN MB" on TABLESPACE_NAME
select file_name,tablespace_name,bytes/1024/1024 "SIZE IN MB" from dba_temp_files
where exists ( select 1 from dba_temp_files
having count(9) > 0 )
order by tablespace_name,file_id
/
spool off
prompt
prompt
set term on
PROMPT OUTPUT SPOOLED TO FREE_SPACE_&&DB_NAME..DOC
Edit FREE_SPACE_&&DB_NAME..DOC
```



Recap

- **LMT – A 30,000 Feet View**
- **So what's wrong with DMT ?**
- **What LMT is not !**
- **LMT – Getting Personal**
- **Tradeoffs**



Questions..?

Bibliography

- Locally Managed Tablespaces by **Jonathan Lewis**
 - www.jlcomp.demon.co.uk
- Locally Managed Tablespaces - Should I Stay Or Should I Go? By Steve Recsky & Andrew Iwanuck, DBCORP Information Systems Inc.
- MetaLink Document 109630.1 - How and where are bitmaps allocated for locally managed tablespaces
- MetaLink Document 105120.1 - Advantages of Using Locally Managed vs Dictionary Managed Tablespaces
- MetaLink Document 123057.1 - Confusing Error Message when Creating a Table on a Local Managed Tablespace
- MetaLink Document 122005.1 - Migration of Dictionary Managed Temporary Tablespace to Locally Managed One
- MetaLink Document 109627.1 - Migrating a Tablespace from Dictionary to Locally Managed Requires Space
- And ofcourse Steve Adams
 - www.ixora.com.au

Contact info



raj_sakthi@yahoo.com



Legal

© Copyright 2003, **Raj Sakthi**. All rights reserved.

Feel free to distribute this, but please give credit for the materials 'borrowed' from this presentation.

Epilogue



- Thanks to **Jason Cohen** for providing this opportunity.
- Thanks to my friends for their support and unflinching endurance for my terrible Jokes..!!
- Last but not the least...

Thank you for Coming.



Enjoy the Night..!!

