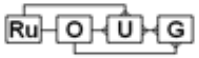
DBA Best Practices

What should I do today?

presentation for:
OUG Harmony 2011



Daniel A. Morgan | **damorgan11g@gmail.com** | **www.morganslibrary.org**

Disclaimer

This room is an unsafe harbour

No one from Oracle has previewed this presentation

No one from Oracle knows what I'm going to say

No one from Oracle knows what I'm going to demo

No one from Oracle has supplied any of my materials

This presentation will include live in SQL*Plus demos because the technology is currently available and works very very well

You may rely upon this presentation to make decisions for your enterprise

This non-disclaimer has not been approved by Oracle Legal

Daniel A. Morgan

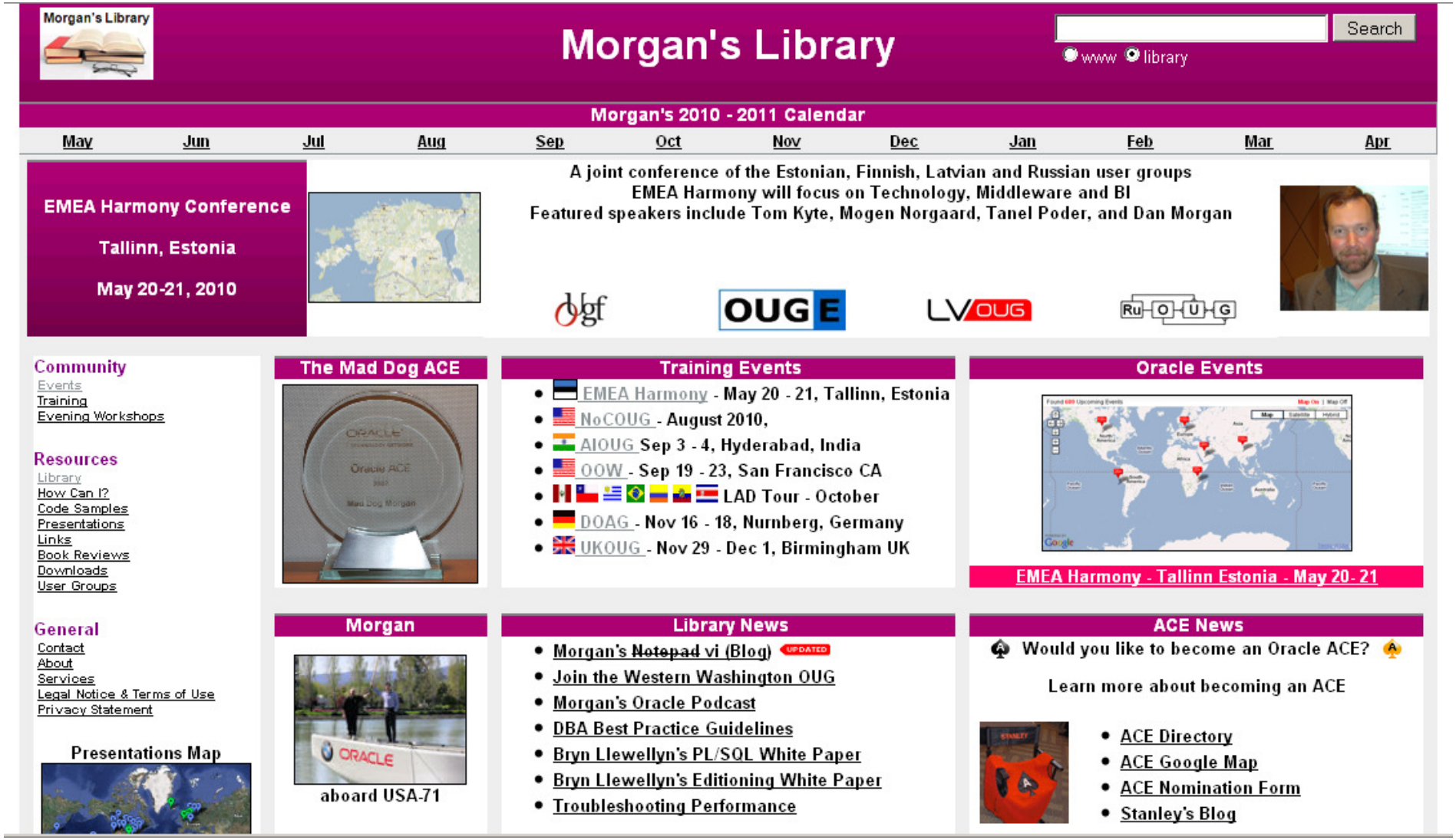
- Oracle ACE Director
- University of Washington Oracle Instructor for 10 years
- The Morgan of Morgan's Library on the web
- Board Member: Western Washington OUG
- Member UKOUG
- Conference Speaker
  - OpenWorld, Collaborate, Kaleidoscope, Brazil, Bulgaria, Canada, Chile, Costa Rica, Denmark, Estonia, Finland, Germany, Japan, New Zealand, Norway, Peru, Sweden, U.K., U.S., Uruguay
- 10g & 11g Beta Tester

cd $MORGAN_HOME

cd $MORGAN_HOME

LAD Tour

Morgan's Library: www.morganslibrary.org



Syllabus

- What should I do today?
- What should I know (about my database)?
- My Oracle Database Bug of the Year

What Should I Do Today?

How do DBAs learn to do their job?

- University
- Books
- Technical Journals
- Oracle Magazine
- Websites

- How do physicians and surgeons know what to do?
- Accountants?
- Engineers?
- Lawyers?
- Pharmacists?

Alert Log Review (1 of 2)

- Keyword search for "ORA-", "Warning" and "Error"

```
Wed Apr 20 13:17:34 2011
Errors in file C:\ORACLE\diag\rdbms\orabase\orabase\trace\orabase_ckpt_13076.trc:
ORA-00206: error in writing (block 3, # blocks 1) of control file
ORA-00202: control file: 'C:\ORACLE\FAST_RECOVERY_AREA\ORABASE\CONTROL02.CTL'
ORA-27072: File I/O error
OSD-04008: WriteFile() failure, unable to write to file
O/S-Error: (OS 33) The process cannot access the file because another process has locked a portion of the file.
Errors in file C:\ORACLE\diag\rdbms\orabase\orabase\trace\orabase_ckpt_13076.trc:
ORA-00221: error on write to control file
ORA-00206: error in writing (block 3, # blocks 1) of control file
ORA-00202: control file: 'C:\ORACLE\FAST_RECOVERY_AREA\ORABASE\CONTROL02.CTL'
ORA-27072: File I/O error
OSD-04008: WriteFile() failure, unable to write to file
O/S-Error: (OS 33) The process cannot access the file because another process has locked a portion of the file.


Tue Jan 18 18:20:18 2011
Create controlfile reuse set database "orabase"
MAXINSTANCES 8
MAXLOGHISTORY 1
MAXLOGFILES 16
MAXLOGMEMBERS 3
MAXDATAFILES 100
Datafile
'C:\oracle\oradata\orabase\SYSTEM01.DBF',
'C:\oracle\oradata\orabase\SYSAUX01.DBF',
'C:\oracle\oradata\orabase\UNDOTBS01.DBF',
'C:\oracle\oradata\orabase\USERS01.DBF'
LOGFILE GROUP 1 ('C:\oracle\oradata\orabase\redo01.log') SIZE 51200K,
GROUP 2 ('C:\oracle\oradata\orabase\redo02.log') SIZE 51200K,
GROUP 3 ('C:\oracle\oradata\orabase\redo03.log') SIZE 51200K RESETLOGS
WARNING: Default Temporary Tablespace not specified in CREATE DATABASE command
Default Temporary Tablespace will be necessary for a locally managed database in future release
```

Altert Log Review (2 of 2)

- Search for Obsolete Startup Parameters

```
Sat May 13 09:17:34 2011
Starting up:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options.
Using parameter settings in server-side spfile C:\ORACLE\PRODUCT\11.2.0\DB_1\DATABASE\SPFILEORABASE.ORA
System parameters with non-default values:
  processes                = 150
  memory_target            = 1232M
  control_files            = "C:\ORACLE\ORADATA\ORABASE\CONTROL01.CTL"
  control_files            = "C:\ORACLE\FLASH_RECOVERY_AREA\ORABASE\CONTROL02.CTL"
  db_block_size            = 8192
  compatible               = "11.2.0.0.0"
  db_recovery_file_dest    = "C:\oracle\flash_recovery_area"
  db_recovery_file_dest_size= 3852M
  undo_tablespace          = "UNDOTBS1"
  remote_login_passwordfile= "EXCLUSIVE"
  db_domain                = ""
  dispatchers              = "(PROTOCOL=TCP) (SERVICE=orabaseXDB)"
  audit_file_dest          = "C:\ORACLE\ADMIN\ORABASE\ADUMP"
  audit_trail              = "DB"
  db_name                  = "orabase"
  open_cursors             = 300
  parallel_automatic_tuning= TRUE
  diagnostic_dest          = "C:\ORACLE"
Deprecated system parameters with specified values:
  parallel_automatic_tuning
End of deprecated system parameter listing
```

Block Level Corruption

- Alert Log
- v$database_block_corruption
- DBMS_DBVERIFY
- DBV Utility

```
SQL> SELECT * FROM v$database_block_corruption where rownum = 1;

      FILE#     BLOCK#     BLOCKS    CORRUPTION_CHANGE# CORRUPTION
---------- ---------- ---------- -------------------- ----------
        22     162844          4       10594075667556 CORRUPT


SQL> select tablespace_name, file_name
  2  from dba_data_files
  3  where file_id = 22;

TABLESPACE_NAME                FILE_NAME
------------------------------ ----------------------------------------
EXAMPLE                        C:\ORACLE\ORADATA\ORABASE\EXAMPLE01.DBF


 SQL> select owner, segment_name
  2  from dba_extents
  3  where file_id = 22
  4  and block_id between 162844 and 162848;

OWNER                          SEGMENT_NAME
------------------------------ ------------
UWCLASS                        DBA_ARGS
```

- **RMAN> RECOVER CORRUPTION LIST;**

Verify Backup Quality

- Do you have (a real) backup?
  - RMAN
  - Snap and Clone
- Can you restore and recover?
  - v$backup_corruption
  - v$copy_corruption
  - **RMAN> BACKUP VALIDATE CHECK LOGICAL;**

Backup Quality Verification

```
SELECT start_time, end_time, status
FROM v$rman_backup_job_details
ORDER BY 1;
```

```
03-FEB-2011 12:15:05 03-FEB-2011 14:05:17 COMPLETED
03-FEB-2011 18:04:46 03-FEB-2011 19:44:57 COMPLETED
04-FEB-2011 00:05:56 FAILED
04-FEB-2011 02:00:54 04-FEB-2011 03:30:21 COMPLETED
04-FEB-2011 08:01:03 04-FEB-2011 09:27:16 COMPLETED
04-FEB-2011 14:09:03 04-FEB-2011 15:09:35 COMPLETED
04-FEB-2011 20:03:17 05-FEB-2011 00:46:34 COMPLETED WITH ERRORS
05-FEB-2011 02:01:54 05-FEB-2011 03:05:00 COMPLETED
05-FEB-2011 08:01:49 05-FEB-2011 09:16:59 COMPLETED
05-FEB-2011 08:36:15 05-FEB-2011 08:37:47 COMPLETED
05-FEB-2011 14:01:35 05-FEB-2011 15:31:33 COMPLETED
05-FEB-2011 20:01:23 05-FEB-2011 22:05:26 COMPLETED WITH ERRORS
06-FEB-2011 02:03:05 06-FEB-2011 03:37:16 COMPLETED
06-FEB-2011 08:02:56 06-FEB-2011 09:24:00 COMPLETED
06-FEB-2011 14:07:53 06-FEB-2011 15:12:10 COMPLETED
06-FEB-2011 20:01:50 06-FEB-2011 21:06:41 FAILED
07-FEB-2011 02:01:42 07-FEB-2011 03:16:17 COMPLETED
07-FEB-2011 08:02:35 07-FEB-2011 09:43:35 COMPLETED
07-FEB-2011 14:03:12 07-FEB-2011 15:14:44 COMPLETED
07-FEB-2011 20:03:56 FAILED
08-FEB-2011 02:04:27 08-FEB-2011 03:24:14 COMPLETED
08-FEB-2011 08:04:43 08-FEB-2011 09:44:53 COMPLETED
08-FEB-2011 08:37:11 08-FEB-2011 08:38:28 COMPLETED
08-FEB-2011 14:05:53 08-FEB-2011 15:52:53 COMPLETED
08-FEB-2011 20:04:20 08-FEB-2011 22:19:13 FAILED
09-FEB-2011 02:05:12 09-FEB-2011 04:15:11 COMPLETED
09-FEB-2011 08:07:40 09-FEB-2011 10:15:32 COMPLETED
09-FEB-2011 14:06:40 09-FEB-2011 15:17:38 COMPLETED
09-FEB-2011 20:05:35 09-FEB-2011 21:47:53 FAILED
10-FEB-2011 02:06:13 10-FEB-2011 03:46:37 COMPLETED
10-FEB-2011 08:06:56 10-FEB-2011 09:44:15 COMPLETED
10-FEB-2011 08:44:16 10-FEB-2011 08:45:33 COMPLETED
10-FEB-2011 14:06:14 10-FEB-2011 15:46:15 COMPLETED
10-FEB-2011 20:07:30 10-FEB-2011 21:54:22 FAILED
11-FEB-2011 00:53:58 11-FEB-2011 03:07:22 COMPLETED
11-FEB-2011 06:53:31 11-FEB-2011 08:32:46 COMPLETED
```

Verify Backup Quality

- SPFILE
  - v$backup_spfile_details
- Control file
  - v$backup_controlfile_details
- Password file
- dot ORA files
  - listener.ora
  - sqlnet.ora
  - tnsnames.ora
- ASM
- Clusterware
  - OCR
  - Voting Disk

Invalid Objects

- Invalid PL/SQL Objects

```
SQL> select owner, object_type, object_name
  2  from dba_objects
  3  where status = 'INVALID'
  4  order by 1,2,3;

OWNER                          OBJECT_TYPE         OBJECT_NAME
------------------------------ ------------------- --------------------
GLOBAL                         VIEW                CHANNEL_DIMVIEW
GLOBAL                         VIEW                CUSTOMER_DIMVIEW
GLOBAL                         VIEW                PRODUCT_DIMVIEW
GLOBAL                         VIEW                TIME_DIMVIEW
GLOBAL                         VIEW                UNITS_CUBE_CUBEVIEW
MLIB                           PROCEDURE           ALLCBLAST
MLIB                           PROCEDURE           ALLEBLAST
MLIB                           PROCEDURE           CBLAST
MLIB                           PROCEDURE           CLEAN_LIST
MLIB                           PROCEDURE           EBLAST
MLIB                           PROCEDURE           VANCBLAST
MLIB                           PROCEDURE           VANEBLAST
UWCLASS                        PACKAGE BODY        INDEX_UTIL
UWCLASS                        PACKAGE BODY        UTL_SPADV


SQL> alter procedure mlib.allcblast compile;
Warning: Procedure altered with compilation errors.

SQL> sho err
Errors for PROCEDURE MLIB.ALLCBLAST:

LINE/COL     ERROR
---------    -----------------------------------------------------
26/7         PL/SQL: Statement ignored
26/7         PLS-00201: identifier 'SYS.DBMS_LOCK' must be declared
```

Invalid Objects

- Invalid Indexes

```
SQL> select owner, index_name, index_type
  2  from dba_indexes
  3  where status not in ('VALID', 'N/A')
  4  order by 1,3,2;

OWNER                          INDEX_NAME                     INDEX_TYPE
------------------------------ ------------------------------ ----------
GSM_DH                         ERC_PATH_HR_IDX                NORMAL
GSM_DH                         NOK_PATH_HR_IDX                NORMAL
GSM_DH                         NOR_PATH_HR_IDX                NORMAL
GSM_DH                         SIE_PATH_HR_IDX                NORMAL
KC4983                         LOCATION_SPATIAL_IDX           DOMAIN
KC4983                         SYS_C0021882                   NORMAL
UMTS_DH                        ERC_BBP_NODEB_KPI_IDX          NORMAL
UMTS_DH                        ERC_CELL_1030_IDX              NORMAL
UMTS_DH                        ERC_CELL_DH_IDX                NORMAL
UMTS_DH                        ERC_CELL_KPI_IDX_BACKUP        NORMAL
UMTS_DH                        ERC_SECTOR_1030_IDX            NORMAL
UMTS_DH                        ERC_SECTOR_DH_IDX              NORMAL
UMTS_DH                        ERC_SECTOR_KPI_IDX_BACKUP      NORMAL
```

Invalid Objects

- Objects in debug mode
  - Thanks to Toad, PLSQL Developer, and similar tools

```
SQL> select owner, type, name
  2  from dba_plsql_object_settings
  3  where plsql_debug = 'TRUE'
  4  order by 1,2,3;

OWNER                          TYPE         NAME
------------------------------ ------------ ------------------------------
E911_NATIONAL_REPORTING        TYPE         TYP_HANDSET_INTRADO
E911_NATIONAL_REPORTING        TYPE         TYP_HANDSET_TCS
E911_NATIONAL_REPORTING        TYPE BODY    TYP_HANDSET_INTRADO
E911_NATIONAL_REPORTING        TYPE BODY    TYP_HANDSET_TCS
JW430N                         PROCEDURE    CHECKERICMAPPING
JW430N                         PROCEDURE    CREATE_ERIC_MAPPING
KG449W                         PACKAGE      BULK_GATHER_V2
KG449W                         PACKAGE      BULK_GATHER_V2_STAGE
KG449W                         PACKAGE      CELLTRAFFICDATA
KG449W                         PACKAGE      LTE_BULK_ETL
KG449W                         PACKAGE      QUERY_CODE_PKG
KG449W                         PACKAGE BODY BULK_GATHER_V2
KG449W                         PACKAGE BODY BULK_GATHER_V2_STAGE
KG449W                         PACKAGE BODY CELLTRAFFICDATA
KG449W                         PACKAGE BODY LTE_BULK_ETL
KG449W                         PACKAGE BODY QUERY_CODE_PKG
KG449W                         PROCEDURE    CRC
KG449W                         PROCEDURE    GEN_CELLTRAFFICDATA
KG449W                         PROCEDURE    GEN_CONFIG_PARAM
KG449W                         PROCEDURE    NOR_DEL
KG449W                         PROCEDURE    VENDORS
M62085                         FUNCTION     ADDARR
```

Invalid Objects

- Objects compiled with the wrong optimizer mode

```
SQL> select owner, type, name, plsql_optimize_level
  2  from dba_plsql_object_settings
  3  where plsql_optimize_level is not null
  4  and plsql_optimize_level <> 2
  5  order by 1,2,3;

OWNER                     TYPE         NAME                           PLSQL_OPTIMIZE_LEVEL
------------------------- ------------ ------------------------------ --------------------
AIRCOM                    TYPE         ALARM_VIEW_ALL_SC_TAB                             1
AIRCOM                    TYPE         ALARM_VIEW_ALL_T                                  1
GLOBAL                    PROCEDURE    CUSTOM_GRANT                                      1
GLOBAL                    PROCEDURE    REVOKE_GRANT                                      1
KG449W                    PACKAGE      BULK_GATHER_V2                                    1
KG449W                    PACKAGE      BULK_GATHER_V2_STAGE                              1
KG449W                    PACKAGE      CELLTRAFFICDATA                                   1
KG449W                    PACKAGE      LTE_BULK_ETL                                      1
KG449W                    PROCEDURE    VENDORS                                           1
SYS                       TYPE         SYS_YOID0000009056$                               1
```

Hung Resumable Sessions

- Do you have resumable transactions enabled?
  - Max Extents Reached
  - No More Space
  - Space Quota Exceeded

Object Type
Cluster
Index and Index [sub]Partition
LOB Partition and Segment
Table and Table [sub]Partition
Tablespace
Temp Segment
Undo Segment

```
SQL> SELECT user_id, session_id, status, timeout, name
  2  FROM dba_resumable;

   USER_ID SESSION_ID STATUS       TIMEOUT NAME
---------- ---------- --------- ---------- ----------------------------------------
        90        150 NORMAL          1800 User UWCLASS(90), Session 150, Instance 1


 SQL> SELECT user_id, session_id, status, timeout, start_time, suspend_time, error_number
  2  FROM dba_resumable;

   USER_ID SESSION_ID STATUS    TIMEOUT START_TIME        SUSPEND_TIME      ERROR_NUMBER
---------- ---------- --------- ------- ----------------- ----------------- ------------
        90        150 SUSPENDED    1800 05/07/11 11:48:02 05/07/11 11:48:09         1653


SQL> SELECT user_id, session_id, status, suspend_time, sql_text
  2  FROM dba_resumable;

USER_ID STATUS    SUSPEND_TIME      SQL_TEXT
------- --------- ----------------- --------------------------------------------------------------------------------
     90 SUSPENDED 05/07/11 11:48:09 INSERT INTO FILL_SPACE (FIDCOL, STRCOL) VALUES (:B1 , RPAD('X', 999, 'X'))
```

Blocked Sessions

- Identify blocking sessions and determine how long they have been blocked

```
SQL> SELECT (
  2    SELECT username
  3    FROM gv$session
  4    WHERE sid=a.sid) blocker,
  5    a.sid, ' is blocking ', (
  6    SELECT username
  7    FROM gv$session
  8    WHERE sid=b.sid) blockee, b.sid
  9  FROM gv$lock a, gv$lock b
 10  WHERE a.block = 1
 11  AND b.request > 0
 12  AND a.id1 = b.id1
 13  AND a.id2 = b.id2;

BLOCKER                          SID 'ISBLOCKING'  BLOCKEE                          SID
------------------------------ ---------- ------------- ------------------------------ ----------
TEST_LOADER_PROC                   717  is blocking  TEST_LOADER_PROC                 252
TEST_LOADER_PROC                   717  is blocking  TEST_LOADER_PROC                1722
```

Surprise Restarts

- With RAC I am too often amazed to find that a node (instance) rebooted itself spontaneously, rejoined the cluster, and no one notice

```
SQL> select inst_id, startup_time
  2  from gv$instance
  3* order by 1;

   INST_ID STARTUP_TIME
---------- --------------------
         5 09-MAY-2011 21:32:36
         6 06-MAY-2011 22:45:15
         7 06-MAY-2011 22:50:19
         8 12-MAY-2011 19:23:57
         9 06-MAY-2011 23:02:11
        10 06-MAY-2011 23:07:50
```

Log Switch Frequency (1 of 2)

```
SELECT TO_CHAR(first_time,'MMDD') MMDD,
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'00',1,0)),'99') "00",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'01',1,0)),'99') "01",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'02',1,0)),'99') "02",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'03',1,0)),'99') "03",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'04',1,0)),'99') "04",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'05',1,0)),'99') "05",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'06',1,0)),'99') "06",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'07',1,0)),'99') "07",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'08',1,0)),'99') "08",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'09',1,0)),'99') "09",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'10',1,0)),'99') "10",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'11',1,0)),'99') "11",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'12',1,0)),'99') "12",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'13',1,0)),'99') "13",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'14',1,0)),'99') "14",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'15',1,0)),'99') "15",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'16',1,0)),'99') "16",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'17',1,0)),'99') "17",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'18',1,0)),'99') "18",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'19',1,0)),'99') "19",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'20',1,0)),'99') "20",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'21',1,0)),'99') "21",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'22',1,0)),'99') "22",
TO_CHAR(SUM(DECODE(TO_CHAR(first_time,'HH24'),'23',1,0)),'99') "23"
FROM v$log_history
GROUP BY TO_CHAR(first_time,'MMDD')
ORDER BY 1;
```

Log Switch Frequency (2 of 2)

MMDD	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0609	16	11	9	8	8	10	12	8	8	10	8	10	14	10	11	15	15	8	12	8	7	6	9	7
0610	13	12	8	9	7	6	11	9	6	8	7	8	12	6	7	6	8	7	10	7	4	4	4	5
0611	12	8	5	9	9	7	11	7	6	7	8	5	12	9	10	8	9	12	12	10	6	6	9	8
0612	13	12	7	9	7	9	10	10	7	7	9	8	11	7	7	8	7	7	11	9	5	6	8	7
0613	12	11	7	8	8	7	13	7	9	7	8	7	13	10	9	8	8	8	11	8	7	5	7	6
0614	15	10	9	9	8	9	13	9	9	7	11	13	11	9	8	9	13	9	12	9	7	9	7	7
0615	15	10	10	8	10	9	12	8	9	8	9	7	13	6	8	7	7	7	15	10	7	7	7	5
0616	13	8	8	7	7	6	10	8	11	7	8	6	11	7	12	13	13	14	13	9	9	9	7	8
0617	15	13	10	9	8	9	16	8	8	10	9	10	16	11	10	10	8	11	13	8	9	9	7	9
0618	12	13	15	15	13	13	15	13	9	12	8	11	14	9	10	9	9	8	14	9	8	8	9	8
0619	16	11	10	11	9	9	13	12	10	9	12	12	17	8	9	9	11	11	14	9	9	11	10	12
0620	19	15	11	10	10	10	19	11	9	9	9	9	13	7	15	10	11	11	12	10	9	11	11	10
0621	13	16	11	9	10	13	16	8	14	9	11	12	17	10	10	11	8	11	14	8	11	14	8	11
0622	16	13	13	11	11	9	16	9	9	11	10	11	17	10	9	10	10	10	13	14	9	10	10	8
0623	19	13	12	13	13	11	16	12	11	11	11	11	16	9	10	13	2	14	14	8	9	8	8	8
0624	14	9	9	9	7	9	11	8	8	7	8	8	14	7	8	7	9	3	6	0	0	0	0	0

0630	7	4	23	19	9	10	5	6	7	17	19	17	15	17	15	43	40	32	17	15	14	20	13	15
0701	15	12	14	12	13	12	13	17	15	17	20	20	18	18	17	15	14	13	10	10	15	15	13	19
0702	21	22	20	18	14	14	12	13	11	11	14	14	14	10	9	10	9	10	11	9	11	9	10	12
0703	9	13	10	17	14	17	15	17	23	20	19	20	17	19	16	17	15	17	15	15	15	16	16	18
0704	22	19	19	18	16	15	13	13	14	11	13	10	12	14	10	12	14	11	9	11	12	13	12	9
0705	14	13	9	11	10	12	13	11	11	8	10	10	11	11	11	12	10	10	9	10	8	9	12	7
0706	14	15	11	12	9	15	13	12	12	9	12	14	12	12	12	12	13	11	8	9	12	13	2	0
0707	0	0	1	0	3	15	10	10	7	8	10	11	12	8	6	9	13	12	9	8	9	8	10	10
0708	16	9	8	15	10	11	9	8	8	14	9	10	10	8	8	14	15	10	9	9	8	9	10	10
0709	13	12	9	10	10	9	9	10	11	11	8	9	9	8	9	13	8	9	6	9	9	11	10	9
0710	12	10	9	10	9	12	9	8	8	11	7	10	11	9	9	13	10	9	8	9	11	12	10	10

What Should I Know?

... about my database

Actual Version (including patches)

```
SQL> DECLARE
  2  ver    VARCHAR2(100);
  3  compat VARCHAR2(100);
  4  BEGIN
  5    dbms_utility.db_version(ver, compat);
  6    dbms_output.put_line('Version: ' || ver ||' Compatible: ' || compat);
  7  END;
  8  /
Version: 11.2.0.2.0 Compatible: 11.2.0.0.0

PL/SQL procedure successfully completed.
```

Installed Components

```
SQL> col comp_name format a40
SQL> SELECT comp_name, version, status
  2  FROM dba_registry;

COMP_NAME                                VERSION                        STATUS
---------------------------------------- ------------------------------ -----------
OWB                                      11.2.0.2.0                     VALID
Oracle Application Express               3.2.1.00.11                    VALID
Oracle Enterprise Manager                11.2.0.2.0                     VALID
OLAP Catalog                             11.2.0.2.0                     VALID
Spatial                                  11.2.0.2.0                     VALID
Oracle Multimedia                        11.2.0.2.0                     VALID
Oracle XML Database                      11.2.0.2.0                     VALID
Oracle Text                              11.2.0.2.0                     VALID
Oracle Expression Filter                 11.2.0.2.0                     VALID
Oracle Rules Manager                     11.2.0.2.0                     VALID
Oracle Workspace Manager                 11.2.0.2.0                     VALID
Oracle Database Catalog Views            11.2.0.2.0                     VALID
Oracle Database Packages and Types       11.2.0.2.0                     VALID
JServer JAVA Virtual Machine             11.2.0.2.0                     VALID
Oracle XDK                               11.2.0.2.0                     VALID
Oracle Database Java Packages            11.2.0.2.0                     VALID
OLAP Analytic Workspace                  11.2.0.2.0                     VALID
Oracle OLAP API                          11.2.0.2.0                     VALID
```

High Availability Options

- RAC
- Data Guard

```
SQL> select inst_id, instance_name, startup_time, status
  2  from gv$instance
  3  order by 1;

   INST_ID INSTANCE_NAME    STARTUP_TIME         STATUS
---------- ---------------- -------------------- ------------
         5 opm01p5          09-MAY-2011 21:32:36 OPEN
         6 opm01p6          06-MAY-2011 22:45:15 OPEN
         7 opm01p7          06-MAY-2011 22:50:19 OPEN
         8 opm01p8          12-MAY-2011 19:23:57 OPEN
         9 opm01p9          06-MAY-2011 23:02:11 OPEN
        10 opm01p10         06-MAY-2011 23:07:50 OPEN


SQL> SELECT protection_mode, protection_level, remote_archive, database_role,
     dataguard_broker, guard_status
  2  FROM v$database;

PROTECTION_MODE      PROTECTION_LEVEL     REMOTE_A DATABASE_ROLE    DATAGUAR GUARD_S
-------------------- -------------------- -------- ---------------- -------- -------
MAXIMUM PERFORMANCE  UNPROTECTED          ENABLED  PRIMARY          DISABLED NONE
```

Other Options

- Feature Usage Report

```
spool c:\temp\usage_report.html

SELECT output
FROM TABLE(dbms_feature_usage_report.display_html);

spool off



spool c:\temp\usage_report.txt

SELECT output
FROM TABLE(dbms_feature_usage_report.display_text);

spool off
```

Flashback Logging Enabled

```
SQL> SELECT flashback_on FROM v$database;


FLASHBACK_ON
------------------
NO
```

Supplemental and Force Logging

```
SQL> SELECT force_logging, supplemental_log_data_min,
  2  supplemental_log_data_pk, supplemental_log_data_ui,
  3  supplemental_log_data_fk, supplemental_log_data_all
  4* FROM v$database;

FOR SUPPLEME SUP SUP SUP SUP
--- -------- --- --- --- ---
NO  NO       NO  NO  NO  NO
```

Control File Location

- Are control files in different physical locations?

```
SQL> SELECT name FROM v$controlfile;

NAME
------------------------------
C:\ORACLE\ORADATA\ORABASE\CONTROL01.CTL

C:\ORACLE\FAST_RECOVERY_AREA\ORABASE\CONTROL02.CTL

D:\ORACLE\BACKUP_FILES\ORABASE\CONTROL03.CTL
```

Redo Log Location and Multiplexing

- Are log files multiplexed to different physical locations?

```
SQL> SELECT l.group#, lf.type, lf.member, l.bytes, l.status LOG_STATUS
  2  FROM gv$log l, gv$logfile lf
  3  WHERE l.group# = lf.group#
  4  AND l.inst_id = lf.inst_id
  5  ORDER BY 1,3;

GROUP# TYPE    MEMBER                                                       BYTES LOG_STAT
---------- ------- ------------------------------------------------------------ ---------- --------
       501 ONLINE  +OPMREDO_A/opm01p/onlinelog/group_501.256.734227433     8388608000 INACTIVE
       501 ONLINE  +OPMREDO_A/opm01p/onlinelog/group_501.256.734227433     8388608000 INACTIVE
       501 ONLINE  +OPMREDO_A/opm01p/onlinelog/group_501.256.734227433     8388608000 INACTIVE
       501 ONLINE  +OPMREDO_A/opm01p/onlinelog/group_501.256.734227433     8388608000 INACTIVE
       501 ONLINE  +OPMREDO_A/opm01p/onlinelog/group_501.256.734227433     8388608000 INACTIVE
       501 ONLINE  +OPMREDO_A/opm01p/onlinelog/group_501.256.734227433     8388608000 INACTIVE
       501 ONLINE  +OPMREDO_B/opm01p/onlinelog/group_501.256.734228593     8388608000 INACTIVE
       501 ONLINE  +OPMREDO_B/opm01p/onlinelog/group_501.256.734228593     8388608000 INACTIVE
       501 ONLINE  +OPMREDO_B/opm01p/onlinelog/group_501.256.734228593     8388608000 INACTIVE
       501 ONLINE  +OPMREDO_B/opm01p/onlinelog/group_501.256.734228593     8388608000 INACTIVE
       501 ONLINE  +OPMREDO_B/opm01p/onlinelog/group_501.256.734228593     8388608000 INACTIVE
       501 ONLINE  +OPMREDO_B/opm01p/onlinelog/group_501.256.734228593     8388608000 INACTIVE
       502 ONLINE  +OPMREDO_A/opm01p/onlinelog/group_502.257.734230275     8388608000 CURRENT
       502 ONLINE  +OPMREDO_A/opm01p/onlinelog/group_502.257.734230275     8388608000 CURRENT
       502 ONLINE  +OPMREDO_A/opm01p/onlinelog/group_502.257.734230275     8388608000 CURRENT
       502 ONLINE  +OPMREDO_A/opm01p/onlinelog/group_502.257.734230275     8388608000 CURRENT
       502 ONLINE  +OPMREDO_A/opm01p/onlinelog/group_502.257.734230275     8388608000 CURRENT
       502 ONLINE  +OPMREDO_A/opm01p/onlinelog/group_502.257.734230275     8388608000 CURRENT
       502 ONLINE  +OPMREDO_B/opm01p/onlinelog/group_502.257.734230645     8388608000 CURRENT
       502 ONLINE  +OPMREDO_B/opm01p/onlinelog/group_502.257.734230645     8388608000 CURRENT
       502 ONLINE  +OPMREDO_B/opm01p/onlinelog/group_502.257.734230645     8388608000 CURRENT
       502 ONLINE  +OPMREDO_B/opm01p/onlinelog/group_502.257.734230645     8388608000 CURRENT
       502 ONLINE  +OPMREDO_B/opm01p/onlinelog/group_502.257.734230645     8388608000 CURRENT
       502 ONLINE  +OPMREDO_B/opm01p/onlinelog/group_502.257.734230645     8388608000 CURRENT
...
     1008 ONLINE  +OPMREDO_B/opm01p/onlinelog/group_1008.303.743188059    8388608000 INACTIVE

576 rows selected.
```

Startup Parameters

```
CREATE PFILE='c:\oracle\admin\orabase\pfile\initparams.txt' FROM memory;
```

```
# Oracle init.ora parameter file generated by instance orabase on 05/08/2011 19:17:46
__db_cache_size=192M
__java_pool_size=8M
__large_pool_size=8M
__oracle_base='C:\oracle' # ORACLE_BASE set from environment
__pga_aggregate_target=488M
__sga_target=744M
__shared_io_pool_size=112M
__shared_pool_size=408M
__streams_pool_size=8M
_aggregation_optimization_settings=0
_always_anti_join='CHOOSE'
_always_semi_join='CHOOSE'
_and_pruning_enabled=TRUE
_b_tree_bitmap_plans=TRUE
_bloom_filter_enabled=TRUE
_bloom_folding_enabled=TRUE
_bloom_pruning_enabled=TRUE
_complex_view_merging=TRUE
_compression_compatibility='11.2.0.0.0'
_connect_by_use_union_all='TRUE'
_convert_set_to_join=FALSE
_cost_equality_semi_join=TRUE
_cpu_to_io=0
_dimension_skip_null=TRUE
_eliminate_common_subexpr=TRUE
_enable_type_dep_selectivity=TRUE
_fast_full_scan_enabled=TRUE
_first_k_rows_dynamic_proration=TRUE
_gby_hash_aggregation_enabled=TRUE
_generalized_pruning_enabled=TRUE
_globalindex_pnum_filter_enabled=TRUE
_gs_anti_semi_join_allowed=TRUE
_improved_outerjoin_card=TRUE
_improved_row_length_enabled=TRUE
_index_join_enabled=TRUE
```

Default Passwords

- Are any users configured with default passwords?

```
SELECT d.username, u.account_status
 FROM dba_users_with_defpwd d, dba_users u
 WHERE d.username = u.username
 AND account_status = 'OPEN'
 ORDER BY 2,1;
```

Block Change Tracking

- Is block change tracking enabled?

```
SQL> SELECT filename, status, bytes
  2  FROM v$block_change_tracking;

FILENAME                                                 STATUS           BYTES
-------------------------------------------------------- ---------- ----------
                                                         DISABLED


SQL> ALTER DATABASE ENABLE BLOCK CHANGE TRACKING
  2  USING FILE 'c:\oracle\fast_recovery_area\ORABASE\bctf01.log';

Database altered.


SQL> SELECT filename, status, bytes
  2  FROM v$block_change_tracking;

FILENAME                                                 STATUS           BYTES
-------------------------------------------------------- ---------- ----------
C:\ORACLE\FAST_RECOVERY_AREA\ORABASE\BCTF01.LOG           ENABLED       11599872
```

Profile Limits

```
SQL> SELECT name, value
  2  FROM gv$parameter
  3  WHERE name = 'resource_limit';

NAME                           VALUE
------------------------------ ------------------------------
resource_limit                 FALSE

SQL> SELECT profile, resource_name, limit
  2  FROM dba_profiles
  3  ORDER BY 1,2;

PROFILE                        RESOURCE_NAME                  LIMIT
------------------------------ ------------------------------ ---------
DEFAULT                        COMPOSITE_LIMIT                UNLIMITED
DEFAULT                        CONNECT_TIME                   UNLIMITED
DEFAULT                        CPU_PER_CALL                   UNLIMITED
DEFAULT                        CPU_PER_SESSION                UNLIMITED
DEFAULT                        FAILED_LOGIN_ATTEMPTS          10
DEFAULT                        IDLE_TIME                      UNLIMITED
DEFAULT                        LOGICAL_READS_PER_CALL         UNLIMITED
DEFAULT                        LOGICAL_READS_PER_SESSION      UNLIMITED
DEFAULT                        PASSWORD_GRACE_TIME            7
DEFAULT                        PASSWORD_LIFE_TIME             180
DEFAULT                        PASSWORD_LOCK_TIME             1
DEFAULT                        PASSWORD_REUSE_MAX             UNLIMITED
DEFAULT                        PASSWORD_REUSE_TIME            UNLIMITED
DEFAULT                        PASSWORD_VERIFY_FUNCTION       NULL
DEFAULT                        PRIVATE_SGA                    UNLIMITED
```

Event Triggers (1 of 2)

- Are event triggers enabled?

```
SQL> SELECT a.obj#, a.sys_evts, b.name, a.enabled
  2  FROM trigger$ a, obj$ b
  3  WHERE a.sys_evts > 0
  4  AND a.obj#=b.obj#
  5  AND baseobject IN (0, 88);

      OBJ#   SYS_EVTS NAME                           ENABLED
---------- ---------- ------------------------------ -------
      8983        128 AW_DROP_TRG                          1
     13045       4096 AW_TRUNC_TRG                         1
     13047       8192 AW_REN_TRG                           1
     13747       8416 NO_VM_DDL                            0
     13748        128 NO_VM_DROP_A                         0
     54852         64 CDC_ALTER_CTABLE_BEFORE              0
     54853         32 CDC_CREATE_CTABLE_AFTER              0
     54854         32 CDC_CREATE_CTABLE_BEFORE             0
     54855        128 CDC_DROP_CTABLE_BEFORE               0
     55754         96 EXPFIL_RESTRICT_TYPEEVOLVE           0
     55755       8256 EXPFIL_ALTEREXPTAB_MAINT             0
     56976       4224 XDB_PI_TRIG                          1
     55752        128 EXPFIL_DROPOBJ_MAINT                 0
     55753        128 EXPFIL_DROPUSR_MAINT                 0
     57745       4096 RLMGR_TRUNCATE_MAINT                 0
     62859         32 SDO_ST_SYN_CREATE                    1
     66945          4 SDO_GEOR_ERR_TRIGGER                 1
     62695        128 SDO_TOPO_DROP_FTBL                   1
     67022        128 SDO_NETWORK_DROP_USER                1
     70492          1 MGMT_STARTUP                         1
     79850          8 LOGON_TRIGGER                        1
```

Event Triggers (2 of 2)

- If so what are they doing?

```
SQL> select trigger_body
  2  from dba_triggers
  3  where trigger_name = 'MGMT_STARTUP';

TRIGGER_BODY
--------------------------------------------------------------------------------
BEGIN
  -- Pin all PLSQL in the SYSMAN schema in memory
  EMD_MAINTENANCE.PIN_PLSQL;
EXCEPTION
  -- Trap all exceptions so database startup will not be impacted
  -- by errors.  Called routines should handle errors appropriately.
  WHEN OTHERS THEN
    NULL;
END;
```

RDA

- If you do not have a current RDA you are at risk of wasting hours of your time
- If you have not downloaded the RDA tool you are at risk of having a prolonged outage
- If you have no idea what I'm talking about your job is at risk

My Oracle Database Bug of the Year

Define Large

- The size of the storage footprint?
- The size of the queries?
- The size of the DML statements?
- The number of transactions per second?
- The number of active sessions?

WORKLOAD REPOSITORY report for

DB Name	DB Id	Instance	Inst num	Startup Time	Release	RAC
OPM01P	782247420	opm01p6	6	18-Aug-10 21:08	11.1.0.7.0	YES

Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)
usp9004b	Solaris[tm] OE (64-bit)	128	64	16	503.16

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	7037	15-Sep-10 13:00:18	406	7.5
End Snap:	7038	15-Sep-10 14:01:28	318	8.5
Elapsed:		61.17 (mins)		
DB Time:		6,076.88 (mins)		

Define Large

SQL ordered by Executions

- Total Executions: 29,717,627
- Captured SQL account for 77.4% of Total

Executions	Rows Processed	Rows per Exec	CPU per Exec (s)	Elap per Exec (s)	SQL Id	SQL Module	SQL Text
10,128,178	2,506,529	0.25	0.00	0.00	932srzg1krc33	ASN_07B_DIP(004110016)	SELECT NE_TIMEZONE FROM CMPM.E...
7,576,759	7,579,197	1.00	0.00	0.00	1h698sb62un99	asci_56_RANAPProtocolStats(016110008)	SELECT DISTINCT NE_TIMEZONE FR...
3,914,621	3,848,268	0.98	0.00	0.00	5tbzddgguu8cc	asci_56_RANAPProtocolStats(016110008)	SELECT SYS_VERSION FROM CMPM.T...
311,645	311,604	1.00	0.00	0.00	7gtztzv329wg0		select c.name, u.name from co...
301,428	301,325	1.00	0.00	0.00	36s446f9cnwhw		SELECT C.NAME FROM COL$ C WHER...
200,692	200,669	1.00	0.00	0.00	4vs91dcv7u1p6	OMS	insert into sys.aud$(sessioni...
65,044	65,035	1.00	0.00	0.00	fz9xwpt2cvt0k		SELECT par_type, param_clob, ...
64,949	3,945,482	60.75	0.00	0.00	f5ra7dru5fk5n	XML_P7R_RNC_RCS(003110008)	SELECT NAME, PATH, READ, WR...
64,801	64,807	1.00	0.00	0.00	fhzj09a7fnrnb	XML_V7I_IN_LP_DC(00811000V)	SELECT DBTIMEZONE, LENGTH(DBT...
64,632	64,542	1.00	0.00	0.00	15jnrrb6016nd	XML_V7I_IN_LP_DC(00811000V)	SELECT SESSIONTIMEZONE, LENGT...

Some of us are not afraid to show ours in public

This Is One Query (page 1 of 5)

Complete List of SQL Text

SQL Id	SQL Text
0cjpvvqy854kh	SELECT 'ERICSSON_CSCORE.EOS' AS TABLENAME, sumday, CASE WHEN percent(summarised_ontime_count, (ontime_count-late_count), 2) > 100 THEN 100 ELSE percent(summarised_ontime_count, (ontime_count-late_count), 2) END percent_of_cutoff, case when 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) < 0 then 0 else 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end percent_of_cutoff_not_summed, percent(summarised_ontime_count, ontime_count+late_count, 2)percentage_of_total_by5am, percent(summarised_ontime_count+summarised_late_count, ontime_count+late_count, 2) current_percentage, ontime_count+late_count current_raw_elements, summarised_ontime_count+summarised_late_count current_summary_elements, summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM, late_count late_raw_elements, ontime_count-late_count available_at_cutoff, percent(num_of_sumrows, num_of_rows, 2) accuracy from (select count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+3/24 then MSC end) late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+3/24 then MSC end) ontime_count, count (*) num_of_rows from ERICSSON_CSCORE.EOS where datetime between trunc(sysdate)-1 and trunc(sysdate)-1/24/60)rawt, (select max(datetime)sumday, count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+7/24 then MSC end) summarised_late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(s ysdate)+7/24 then MSC end) summarised_ontime_count, sum(entries) num_of_sumrows from ERICSSON_CSCORE.EOS_DY where datetime = trunc(sysdate)-1)sumt union all SELECT 'ERICSSON_CSCORE.MSCSTATS' AS TABLENAME, sumday, CASE WHEN percent(summarised_ontime_count, (ontime_count-late_count), 2) > 100 THEN 100 ELSE percent(summarised_ontime_count, (ontime_count-late_count), 2) END percent_of_cutoff, case when 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) < 0 then 0 else 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end percent_of_cutoff_not_summed, percent(summarised_ontime_count, ontime_count+late_count, 2)percentage_of_total_by5am, percent(summarised_ontime_count+summarised_late_count, ontime_count+late_count, 2) current_percentage, ontime_count+late_count current_raw_elements, summarised_ontime_count+summarised_late_count current_summary_elements, summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM, late_count late_raw_elements, ontime_count-late_count available_at_cutoff, percent(num_of_sumrows, num_of_rows, 2) accuracy from (select count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+3/24 then MSC end) late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+3/24 then MSC end) ontime_count, count (*) num_of_rows from ERICSSON_CSCORE.MSCSTATS where datetime between trunc(sysdate)-1 and trunc(sysdate)-1/24/60)rawt, (select max(datetime)sumday, count(dist inct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+7/24 then MSC end) summarised_late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+7/24 then MSC end) summarised_ontime_count, sum(entries) num_of_sumrows from ERICSSON_CSCORE.MSCSTATS_DY where datetime = trunc(sysdate)-1)sumt union all SELECT 'ERICSSON_CSCORE.MTRAFTYPE' AS TABLENAME, sumday, CASE WHEN percent(summarised_ontime_count, (ontime_count-late_count), 2) > 100 THEN 100 ELSE percent(summarised_ontime_count, (ontime_count-late_count), 2) END percent_of_cutoff, case when 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) < 0 then 0 else 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end percent_of_cutoff_not_summed, percent(summarised_ontime_count, ontime_count+late_count, 2)percentage_of_total_by5am, percent(summarised_ontime_count+summarised_late_count, ontime_count+late_count, 2) current_percentage, ontime_count+late_count current_raw_elements, summarised_ontime_count+summarised_late_count current_summary_elements, summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM, late_count late_raw_elements, ontime_count-late_count available_at_cutoff, percent(num_of_sumrows, num_of_rows, 2) accuracy from (select count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+3/24 then MSC end) late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+3/24 then MSC end) ontime_count , count (*) num_of_rows from ERICSSON_CSCORE.MTRAFTYPE where datetime between trunc(sysdate)-1 and trunc(sysdate)-1/24/60)rawt, (select max(datetime)sumday, count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+7/24 then MSC end) summarised_late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+7/24 then MSC end) summarised_ontime_count, sum(entries) num_of_sumrows from ERICSSON_CSCORE.MTRAFTYPE_DY where datetime = trunc(sysdate)-1)sumt union all SELECT 'ERICSSON_GERAN.CELLGPRS' AS TABLENAME, sumday, CASE WHEN percent(summarised_ontime_count, (ontime_count-late_count), 2) > 100 THEN 100 ELSE percent(summarised_ontime_count, (ontime_count-late_count), 2) END percent_of_cutoff, case when 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) < 0 then 0 else 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end percent_of_cutoff_not_summed, percent(summarised_ontime_count, ontime_count+late_count, 2)percentage_of_total_by5am, percent(summarised_ontime_count+summarised_late_count, ontime_count+late_count, 2) current_percentage, ontime_count+late_count current_raw_elements, summarised_ontime_count+summarised_late_count current_summary_elements, summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM, late_count late_raw_elements, ontime_count-late_count available_at_cutoff, percent(num_of_sumrows, num_of_rows, 2) accuracy from (select count(distinct case when trunc(d atetime_ins, 'mi') > trunc(sysdate)+3/24 then BSC end) late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+3/24 then BSC end) ontime_count, count (*) num_of_rows from ERICSSON_GERAN.CELLGPRS where datetime between trunc(sysdate)-1 and trunc(sysdate)-1/24/60)rawt, (select max(datetime)sumday, count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+7/24 then BSC end) summarised_late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+7/24 then BSC end) summarised_ontime_count, sum(entries) num_of_sumrows from ERICSSON_GERAN.CELLGPRS_DY where datetime = trunc(sysdate)-1)sumt union all SELECT 'ERICSSON_GERAN.CELLSTATS' AS TABLENAME, sumday, CASE WHEN percent(summarised_ontime_count, (ontime_count-late_count), 2) > 100 THEN 100 ELSE percent(summarised_ontime_count, (ontime_count-late_count), 2) END percent_of_cutoff, case when 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) < 0 then 0 else 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end percent_of_cutoff_not_summed, percent(summarised_ontime_count, ontime_count+late_count, 2)percentage_of_total_by5am, percent(summarised_ontime_count+summarised_late_count, ontime_count+late_count, 2) current_percentage, ontime_count+late_count current_raw_elements, summarised_ontime_count+summarised_late_count current_summary_elements, summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM, late_ count late_raw_elements, ontime_count-late_count available_at_cutoff, percent(num_of_sumrows, num_of_rows, 2) accuracy from (select count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+3/24 then BSC end) late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+3/24 then BSC end) ontime_count, count (*) num_of_rows from ERICSSON_GERAN.CELLSTATS where datetime between trunc(sysdate)-1 and trunc(sysdate)-1/24/60)rawt, (select max(datetime)sumday, count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+7/24 then BSC end) summarised_late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+7/24 then BSC end) summarised_ontime_count, sum(entries) num_of_sumrows from ERICSSON_GERAN.CELLSTATS_DY where datetime = trunc(sysdate)-1)sumt union all SELECT 'ERICSSON_PSCORE.MM' AS TABLENAME, sumday, CASE WHEN percent(summarised_ontime_count, (ontime_count-late_count), 2) > 100 THEN 100 ELSE percent(summarised_ontime_count, (ontime_count-late_count), 2) END percent_of_cutoff, case when 100-percent(summarised_ontime_count,

This Is One Query (page 2 of 5)

```
(ontime_count-late_count), 2) < 0 then 0 else 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end percent_of_cutoff_not_summed, percent(summarised_ontime_count, ontime_count+late_count,
2)percentage_of_total_by5am, percent(summarised_ontime_count+summarised_late_count, ontime_count+late_count, 2) current_percentage, ontime_count+late_count curr ent_raw_elements,
summarised_ontime_count+summarised_late_count current_summary_elements, summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM, late_count late_raw_elements, ontime_count-late_count available_at_cutoff,
percent(num_of_sumrows, num_of_rows, 2) accuracy from ( select count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+3/24 then SGSNID end) late_count, count(distinct case when trunc(datetime_ins, 'mi') <
trunc(sysdate)+3/24 then SGSNID end) ontime_count, count (*) num_of_rows from ERICSSON_PSCORE.MM where datetime between trunc(sysdate)-1 and trunc(sysdate)-1/24/60 )rawt, ( select max(datetime)sunday, count(distinct case
when trunc(datetime_ins, 'mi') > trunc(sysdate)+7/24 then SGSNID end) summarised_late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+7/24 then SGSNID end) summarised_ontime_count, sum(entries)
num_of_sumrows from ERICSSON_PSCORE.MM_DY where datetime = trunc(sysdate)-1 )sumt union all SELECT 'ERICSSON_PSCORE.SGSNSTAT' AS TABLENAME, sunday, CASE WHEN percent(summarised_ontime_count,
(ontime_count-late_count), 2) > 100 THEN 100 ELSE percent(summarised_ontime_count, (ontime_count-late_count), 2) END percent_of_cutoff, case when 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) < 0 then 0
else 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end percent_of_cutoff_not_summed, percent(summarised_ontime_count, ontime_count+la te_count, 2)percentage_of_total_by5am,
percent(summarised_ontime_count+summarised_late_count, ontime_count+late_count, 2) current_percentage, ontime_count+late_count current_raw_elements, summarised_ontime_count+summarised_late_count
current_summary_elements, summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM, late_count late_raw_elements, ontime_count-late_count available_at_cutoff, percent(num_of_sumrows, num_of_rows, 2)
accuracy from ( select count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+3/24 then SGSNID end) late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+3/24 then SGSNID end) ontime_count,
count (*) num_of_rows from ERICSSON_PSCORE.SGSNSTAT where datetime between trunc(sysdate)-1 and trunc(sysdate)-1/24/60 )rawt, ( select max(datetime)sunday, count(distinct case when trunc(datetime_ins, 'mi') >
trunc(sysdate)+7/24 then SGSNID end) summarised_late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+7/24 then SGSNID end) summarised_ontime_count, sum(entries) num_of_sumrows from
ERICSSON_PSCORE.SGSNSTAT_DY where datetime = trunc(sysdate)-1 )sumt union all SELECT 'ERICSSON_UTRAN.RBS_CARRIER' AS TABLENAME, sunday, CASE WHEN percent(summarised_ontime_count, (ontime_count-late_count),
2) > 100 THEN 100 ELSE percent(summarised_ontime_count, (ontime_count-late_count), 2) END percent_of_cutoff, case when 100-percent(summarised_ontime_count, (ontime_count- late_count), 2) < 0 then 0 else
100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end percent_of_cutoff_not_summed, percent(summarised_ontime_count, ontime_count+late_count, 2)percentage_of_total_by5am,
percent(summarised_ontime_count+summarised_late_count, ontime_count+late_count, 2) current_percentage, ontime_count+late_count current_raw_elements, summarised_ontime_count+summarised_late_count
current_summary_elements, summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM, late_count late_raw_elements, ontime_count-late_count available_at_cutoff, percent(num_of_sumrows, num_of_rows, 2)
accuracy from ( select count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+3/24 then NODEB end) late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+3/24 then NODEB end) ontime_count,
count (*) num_of_rows from ERICSSON_UTRAN.RBS_CARRIER where datetime between trunc(sysdate)-1 and trunc(sysdate)-1/24/60 )rawt, ( select max(datetime)sunday, count(distinct case when trunc(datetime_ins, 'mi') >
trunc(sysdate)+7/24 then NODEB end) summarised_late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+7/24 then NODEB end) summarised_ontime_count, sum(entries) num_of_sumrows from
ERICSSON_UTRAN.RBS_CARRIER_DY where datetime = trunc(sysdate)-1 )sumt union all SELECT 'ERICSSON_UTRAN.RBS_EDCHRES' AS TABLENAME, sunday, CASE WHEN percent(summarised_ontime_count, (ontime_count-l
ate_count), 2) > 100 THEN 100 ELSE percent(summarised_ontime_count, (ontime_count-late_count), 2) END percent_of_cutoff, case when 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) < 0 then 0 else
100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end percent_of_cutoff_not_summed, percent(summarised_ontime_count, ontime_count+late_count, 2)percentage_of_total_by5am,
percent(summarised_ontime_count+summarised_late_count, ontime_count+late_count, 2) current_percentage, ontime_count+late_count current_raw_elements, summarised_ontime_count+summarised_late_count
current_summary_elements, summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM, late_count late_raw_elements, ontime_count-late_count available_at_cutoff, percent(num_of_sumrows, num_of_rows, 2)
accuracy from ( select count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+3/24 then NODEB end) late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+3/24 then NODEB end) ontime_count,
count (*) num_of_rows from ERICSSON_UTRAN.RBS_EDCHRES where datetime between trunc(sysdate)-1 and trunc(sysdate)-1/24/60 )rawt, ( select max(datetime)sunday, count(distinct case when trunc(datetime_ins, 'mi') >
trunc(sysdate)+7/24 then NODEB end) summarised_late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+7/24 then NODEB end) summarised_ontime_count, sum(entries) num_of_sumrows from ERICSSON_UT
RAN.RBS_EDCHRES_DY where datetime = trunc(sysdate)-1 )sumt union all SELECT 'ERICSSON_UTRAN.RBS_HSDSCHRES' AS TABLENAME, sunday, CASE WHEN percent(summarised_ontime_count, (ontime_count-late_count), 2) > 100
THEN 100 ELSE percent(summarised_ontime_count, (ontime_count-late_count), 2) END percent_of_cutoff, case when 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) < 0 then 0 else
100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end percent_of_cutoff_not_summed, percent(summarised_ontime_count, ontime_count+late_count, 2)percentage_of_total_by5am,
percent(summarised_ontime_count+summarised_late_count, ontime_count+late_count, 2) current_percentage, ontime_count+late_count current_raw_elements, summarised_ontime_count+summarised_late_count
current_summary_elements, summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM, late_count late_raw_elements, ontime_count-late_count available_at_cutoff, percent(num_of_sumrows, num_of_rows, 2)
accuracy from ( select count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+3/24 then NODEB end) late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+3/24 then NODEB end) ontime_count,
count (*) num_of_rows from ERICSSON_UTRAN.RBS_HSDSCHRES where datetime between trunc(sysdate)-1 and trunc(sysdate)-1/24/60 )rawt, ( select max(datetime)sunday, count(distinct case when trunc(datetime_ins, 'mi') >
trunc(sysdate)+7/24 then NODEB e nd) summarised_late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+7/24 then NODEB end) summarised_ontime_count, sum(entries) num_of_sumrows from
ERICSSON_UTRAN.RBS_HSDSCHRES_DY where datetime = trunc(sysdate)-1 )sumt union all SELECT 'ERICSSON_UTRAN.RNC_IURLINK' AS TABLENAME, sunday, CASE WHEN percent(summarised_ontime_count, (ontime_count-
late_count), 2) > 100 THEN 100 ELSE percent(summarised_ontime_count, (ontime_count-late_count), 2) END percent_of_cutoff, case when 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) < 0 then 0 else
100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end percent_of_cutoff_not_summed, percent(summarised_ontime_count, ontime_count+late_count, 2)percentage_of_total_by5am,
```

This Is One Query (page 3 of 5)

```
percent(summarised_ontime_count+summarised_late_count, ontime_count+late_count, 2) current_percentage, ontime_count+late_count current_raw_elements, summarised_ontime_count+summarised_late_count
current_summary_elements, summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM, late_count late_raw_elements, ontime_count-late_count available_at_cutoff, percent(num_of_sumrows, num_of_rows, 2)
accuracy from ( select count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+3/24 then RNC end) late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+3/24 then RNC end) ontime_count, count (*)
num_of_rows from ERICSSON_UTRAN.RNC_IURLINK where dateti me between trunc(sysdate)-1 and trunc(sysdate)-1/24/60 )rawt, ( select max(datetime)sunday, count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+7/24
then RNC end) summarised_late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+7/24 then RNC end) summarised_ontime_count, sum(entries) num_of_sumrows from ERICSSON_UTRAN.RNC_IURLINK_DY
where datetime = trunc(sysdate)-1 )sumt union all SELECT 'ERICSSON_UTRAN.RNC_RNCFUNC' AS TABLENAME, sunday, CASE WHEN percent(summarised_ontime_count, (ontime_count-late_count), 2) > 100 THEN 100 ELSE
percent(summarised_ontime_count, (ontime_count-late_count), 2) END percent_of_cutoff, case when 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) < 0 then 0 else 100-percent(summarised_ontime_count,
(ontime_count-late_count), 2) end percent_of_cutoff_not_summed, percent(summarised_ontime_count, ontime_count+late_count, 2)percentage_of_total_by5am, percent(summarised_ontime_count+summarised_late_count,
ontime_count+late_count, 2) current_percentage, ontime_count+late_count current_raw_elements, summarised_ontime_count+summarised_late_count current_summary_elements, summarised_ontime_count-summarised_late_count
SUM_ELEMENTS_AT5AM, late_count late_raw_elements, ontime_count-late_count available_at_cutoff, percent(num_of_sumrows, num_of_rows, 2) accuracy from ( select count(distinct case when trunc(datetime_ins, 'mi') >
trunc(sysdate)+3/24 then RNC end) late_c ount, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+3/24 then RNC end) ontime_count, count (*) num_of_rows from ERICSSON_UTRAN.RNC_RNCFUNC where datetime
between trunc(sysdate)-1 and trunc(sysdate)-1/24/60 )rawt, ( select max(datetime)sunday, count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+7/24 then RNC end) summarised_late_count, count(distinct case when
trunc(datetime_ins, 'mi') < trunc(sysdate)+7/24 then RNC end) summarised_ontime_count, sum(entries) num_of_sumrows from ERICSSON_UTRAN.RNC_RNCFUNC_DY where datetime = trunc(sysdate)-1 )sumt union all SELECT
'ERICSSON_UTRAN.RNC_UCELL' AS TABLENAME, sunday, CASE WHEN percent(summarised_ontime_count, (ontime_count-late_count), 2) > 100 THEN 100 ELSE percent(summarised_ontime_count, (ontime_count-late_count), 2) END
percent_of_cutoff, case when 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) < 0 then 0 else 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end percent_of_cutoff_not_summed,
percent(summarised_ontime_count, ontime_count+late_count, 2)percentage_of_total_by5am, percent(summarised_ontime_count+summarised_late_count, ontime_count+late_count, 2) current_percentage, ontime_count+late_count
current_raw_elements, summarised_ontime_count+summarised_late_count current_summary_elements, summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM, late_count late_raw_elements, ontime_count-
late_count availabl e_at_cutoff, percent(num_of_sumrows, num_of_rows, 2) accuracy from ( select count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+3/24 then RNC end) late_count, count(distinct case when
trunc(datetime_ins, 'mi') < trunc(sysdate)+3/24 then RNC end) ontime_count, count (*) num_of_rows from ERICSSON_UTRAN.RNC_UCELL where datetime between trunc(sysdate)-1 and trunc(sysdate)-1/24/60 )rawt, ( select
max(datetime)sunday, count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+7/24 then RNC end) summarised_late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+7/24 then RNC end)
summarised_ontime_count, sum(entries) num_of_sumrows from ERICSSON_UTRAN.RNC_UCELL_DY where datetime = trunc(sysdate)-1 )sumt union all SELECT 'NORTEL_CSCORE.BASE_OM' AS TABLENAME, sunday, CASE WHEN
percent(summarised_ontime_count, (ontime_count-late_count), 2) > 100 THEN 100 ELSE percent(summarised_ontime_count, (ontime_count-late_count), 2) END percent_of_cutoff, case when 100-percent(summarised_ontime_count,
(ontime_count-late_count), 2) < 0 then 0 else 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end percent_of_cutoff_not_summed, percent(summarised_ontime_count, ontime_count+late_count,
2)percentage_of_total_by5am, percent(summarised_ontime_count+summarised_late_count, ontime_count+late_count, 2) current_percentage, ontime_count+late_count current_raw_elements, summarised_ontime_count+summarised_
late_count current_summary_elements, summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM, late_count late_raw_elements, ontime_count-late_count available_at_cutoff, percent(num_of_sumrows,
num_of_rows, 2) accuracy from ( select count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+3/24 then MSC end) late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+3/24 then MSC end)
ontime_count, count (*) num_of_rows from NORTEL_CSCORE.BASE_OM where datetime between trunc(sysdate)-1 and trunc(sysdate)-1/24/60 )rawt, ( select max(datetime)sunday, count(distinct case when trunc(datetime_ins, 'mi') >
trunc(sysdate)+7/24 then MSC end) summarised_late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+7/24 then MSC end) summarised_ontime_count, sum(entries) num_of_sumrows from
NORTEL_CSCORE.BASE_OM_DY where datetime = trunc(sysdate)-1 )sumt union all SELECT 'NORTEL_CSCORE.C7SP' AS TABLENAME, sunday, CASE WHEN percent(summarised_ontime_count, (ontime_count-late_count), 2) > 100 THEN
100 ELSE percent(summarised_ontime_count, (ontime_count-late_count), 2) END percent_of_cutoff, case when 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) < 0 then 0 else
100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end percent_of_cutoff_not_summed, percent(summarised_ontime_count, ontime_count+late_count, 2)percentage_of_total_by5am, percent(summarised_ontime_c
ount+summarised_late_count, ontime_count+late_count, 2) current_percentage, ontime_count+late_count current_raw_elements, summarised_ontime_count+summarised_late_count current_summary_elements,
summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM, late_count late_raw_elements, ontime_count-late_count available_at_cutoff, percent(num_of_sumrows, num_of_rows, 2) accuracy from ( select
count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+3/24 then MSC end) late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+3/24 then MSC end) ontime_count, count (*) num_of_rows from
NORTEL_CSCORE.C7SP where datetime between trunc(sysdate)-1 and trunc(sysdate)-1/24/60 )rawt, ( select max(datetime)sunday, count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+7/24 then MSC end)
summarised_late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+7/24 then MSC end) summarised_ontime_count, sum(entries) num_of_sumrows from NORTEL_CSCORE.C7SP_DY where datetime =
trunc(sysdate)-1 )sumt union all SELECT 'NORTEL_CSCORE.CCS7' AS TABLENAME, sunday, CASE WHEN percent(summarised_ontime_count, (ontime_count-late_count), 2) > 100 THEN 100 ELSE percent(summarised_ontime_count,
(ontime_count-late_count), 2) END percent_of_cutoff, case when 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) < 0 then 0 else 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end
percent_of_cutoff_not_summed, percent(summarised_ontime_count, ontime_count+late_count, 2)percentage_of_total_by5am, percent(summarised_ontime_count+summarised_late_count, ontime_count+late_count, 2)
current_percentage, ontime_count+late_count current_raw_elements, summarised_ontime_count+summarised_late_count current_summary_elements, summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM,
late_count late_raw_elements, ontime_count-late_count available_at_cutoff, percent(num_of_sumrows, num_of_rows, 2) accuracy from ( select count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+3/24 then MSC end)
```

This Is One Query (page 4 of 5)

```
late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+3/24 then MSC end) ontime_count, count (*) num_of_rows from NORTEL_CSCORE.CCS7 where datetime between trunc(sysdate)-1 and
trunc(sysdate)-1/24/60 )rawt, ( select max(datetime)sunday, count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+7/24 then MSC end) summarised_late_count, count(distinct case when trunc(datetime_ins, 'mi') <
trunc(sysdate)+7/24 then MSC end) summarised_ontime_count, sum(entries) num_of_sumrows from NORTEL_CSCORE.CCS7_DY where datetime = trunc(sysdate)-1 )sumt union all SELECT 'NORTEL_CSCORE.LOCATIONAREACODE' AS
TABLENAME, sunday, CASE WHEN percent(summarised_ontime_count, (ontime_count-late_count), 2) > 100 THEN 100 ELSE percent(summarised_ontime_count, (ontime_count-late_count), 2) END percent_of_cut off, case when
100-percent(summarised_ontime_count, (ontime_count-late_count), 2) < 0 then 0 else 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end percent_of_cutoff_not_summed, percent(summarised_ontime_count,
ontime_count+late_count, 2)percentage_of_total_by5am, percent(summarised_ontime_count+summarised_late_count, ontime_count+late_count, 2) current_percentage, ontime_count+late_count current_raw_elements,
summarised_ontime_count+summarised_late_count current_summary_elements, summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM, late_count late_raw_elements, ontime_count-late_count available_at_cutoff,
percent(num_of_sumrows, num_of_rows, 2) accuracy from ( select count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+3/24 then MSC end) late_count, count(distinct case when trunc(datetime_ins, 'mi') <
trunc(sysdate)+3/24 then MSC end) ontime_count, count (*) num_of_rows from NORTEL_CSCORE.LOCATIONAREACODE where datetime between trunc(sysdate)-1 and trunc(sysdate)-1/24/60 )rawt, ( select max(datetime)sunday,
count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+7/24 then MSC end) summarised_late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+7/24 then MSC end) summarised_ontime_count,
sum(entries) num_of_sumrows from NORTEL_CSCORE.LOCATIONAREACODE_DY where datetime = trunc(sysdate)-1 )sumt union all SELECT 'NORTEL_CSCORE.MSCCP_CP' AS TABLENAME, sund ay, CASE WHEN
percent(summarised_ontime_count, (ontime_count-late_count), 2) > 100 THEN 100 ELSE percent(summarised_ontime_count, (ontime_count-late_count), 2) END percent_of_cutoff, case when 100-percent(summarised_ontime_count,
(ontime_count-late_count), 2) < 0 then 0 else 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end percent_of_cutoff_not_summed, percent(summarised_ontime_count, ontime_count+late_count,
2)percentage_of_total_by5am, percent(summarised_ontime_count+summarised_late_count, ontime_count+late_count, 2) current_percentage, ontime_count+late_count current_raw_elements,
summarised_ontime_count+summarised_late_count current_summary_elements, summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM, late_count late_raw_elements, ontime_count-late_count available_at_cutoff,
percent(num_of_sumrows, num_of_rows, 2) accuracy from ( select count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+3/24 then MSC end) late_count, count(distinct case when trunc(datetime_ins, 'mi') <
trunc(sysdate)+3/24 then MSC end) ontime_count, count (*) num_of_rows from NORTEL_CSCORE.MSCCP_CP where datetime between trunc(sysdate)-1 and trunc(sysdate)-1/24/60 )rawt, ( select max(datetime)sunday, count(distinct
case when trunc(datetime_ins, 'mi') > trunc(sysdate)+7/24 then MSC end) summarised_late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+7/24 then MSC end) summarised_ontime_c ount, sum(entries)
num_of_sumrows from NORTEL_CSCORE.MSCCP_CP_DY where datetime = trunc(sysdate)-1 )sumt union all SELECT 'NORTEL_CSCORE.MSCCP' AS TABLENAME, sunday, CASE WHEN percent(summarised_ontime_count, (ontime_count-
late_count), 2) > 100 THEN 100 ELSE percent(summarised_ontime_count, (ontime_count-late_count), 2) END percent_of_cutoff, case when 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) < 0 then 0 else
100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end percent_of_cutoff_not_summed, percent(summarised_ontime_count, ontime_count+late_count, 2)percentage_of_total_by5am,
percent(summarised_ontime_count+summarised_late_count, ontime_count+late_count, 2) current_percentage, ontime_count+late_count current_raw_elements, summarised_ontime_count+summarised_late_count
current_summary_elements, summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM, late_count late_raw_elements, ontime_count-late_count available_at_cutoff, percent(num_of_sumrows, num_of_rows, 2)
accuracy from ( select count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+3/24 then MSC end) late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+3/24 then MSC end) ontime_count, count (*)
num_of_rows from NORTEL_CSCORE.MSCCP where datetime between trunc(sysdate)-1 and trunc(sysdate)-1/24/60 )rawt, ( select max(datetime)sunday, count(distinct case when trunc(datetime_ins, 'mi') > trunc(s ysdate)+7/24 then
MSC end) summarised_late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+7/24 then MSC end) summarised_ontime_count, sum(entries) num_of_sumrows from NORTEL_CSCORE.MSCCP_DY where datetime =
trunc(sysdate)-1 )sumt union all SELECT 'NORTEL_CSCORE.TRKGRP' AS TABLENAME, sunday, CASE WHEN percent(summarised_ontime_count, (ontime_count-late_count), 2) > 100 THEN 100 ELSE percent(summarised_ontime_count,
(ontime_count-late_count), 2) END percent_of_cutoff, case when 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) < 0 then 0 else 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end
percent_of_cutoff_not_summed, percent(summarised_ontime_count, ontime_count+late_count, 2)percentage_of_total_by5am, percent(summarised_ontime_count+summarised_late_count, ontime_count+late_count, 2)
current_percentage, ontime_count+late_count current_raw_elements, summarised_ontime_count+summarised_late_count current_summary_elements, summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM,
late_count late_raw_elements, ontime_count-late_count available_at_cutoff, percent(num_of_sumrows, num_of_rows, 2) accuracy from ( select count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+3/24 then MSC end)
late_count, count(distinct case when trunc(datetime_ins, 'mi') < trunc(sysdate)+3/24 then MSC end) ontime_count, count (*) num_of_rows from NORTEL_CSCORE.TRKGRP where dateti me between trunc(sysdate)-1 and
trunc(sysdate)-1/24/60 )rawt, ( select max(datetime)sunday, count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+7/24 then MSC end) summarised_late_count, count(distinct case when trunc(datetime_ins, 'mi') <
trunc(sysdate)+7/24 then MSC end) summarised_ontime_count, sum(entries) num_of_sumrows from NORTEL_CSCORE.TRKGRP_DY where datetime = trunc(sysdate)-1 )sumt union all SELECT 'NORTEL_CSCORE.VLR6' AS TABLENAME,
sunday, CASE WHEN percent(summarised_ontime_count, (ontime_count-late_count), 2) > 100 THEN 100 ELSE percent(summarised_ontime_count, (ontime_count-late_count), 2) END percent_of_cutoff, case when
100-percent(summarised_ontime_count, (ontime_count-late_count), 2) < 0 then 0 else 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end percent_of_cutoff_not_summed, percent(summarised_ontime_count,
ontime_count+late_count, 2)percentage_of_total_by5am, percent(summarised_ontime_count+summarised_late_count, ontime_count+late_count, 2) current_percentage, ontime_count+late_count current_raw_elements,
summarised_ontime_count+summarised_late_count current_summary_elements, summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM, late_count late_raw_elements, ontime_count-late_count available_at_cutoff,
percent(num_of_sumrows, num_of_rows, 2) accuracy from ( select count(distinct case when trunc(datetime_ins, 'mi') > trunc(sysdate)+3/24 then MSC end) late_count, count(d istinct case when trunc(datetime_ins, 'mi') <
trunc(sysdate)+3/24 then MSC end) ontime_count, count (*) num_of_rows from NORTEL_CSCORE.VLR6 where datetime between trunc(sysdate)-1 and trunc(sysdate)-1/24/60 )rawt, ( select max(datetime)sunday, count(distinct case
```

This Is One Query (page 5 of 5)

```
when trunc(datetime_ins,'mi') > trunc(sysdate)+7/24 then MSC end) summarised_late_count, count(distinct case when trunc(datetime_ins,'mi') < trunc(sysdate)+7/24 then MSC end) summarised_ontime_count, sum(entries)
num_of_sumrows from NORTEL_CSCORE.VLR6_DY where datetime = trunc(sysdate)-1 )sumt union all SELECT 'NORTEL_PSCORE.GSCGMM' AS TABLENAME, sumday, CASE WHEN percent(summarised_ontime_count, (ontime_count-
late_count), 2) > 100 THEN 100 ELSE percent(summarised_ontime_count, (ontime_count-late_count), 2) END percent_of_cutoff, case when 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) < 0 then 0 else
100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end percent_of_cutoff_not_summed, percent(summarised_ontime_count, ontime_count+late_count, 2)percentage_of_total_by5am,
percent(summarised_ontime_count+summarised_late_count, ontime_count+late_count, 2) current_percentage, ontime_count+late_count current_raw_elements, summarised_ontime_count+summarised_late_count
current_summary_elements, summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM, late_count late_raw_elements, ontime_count-late_count available_at_cutoff, percent(num_of_sumro ws, num_of_rows, 2)
accuracy from ( select count(distinct case when trunc(datetime_ins,'mi') > trunc(sysdate)+3/24 then SGSN end) late_count, count(distinct case when trunc(datetime_ins,'mi') < trunc(sysdate)+3/24 then SGSN end) ontime_count, count
(*) num_of_rows from NORTEL_PSCORE.GSCGMM where datetime between trunc(sysdate)-1 and trunc(sysdate)-1/24/60 )rawt, ( select max(datetime)sumday, count(distinct case when trunc(datetime_ins,'mi') > trunc(sysdate)+7/24
then SGSN end) summarised_late_count, count(distinct case when trunc(datetime_ins,'mi') < trunc(sysdate)+7/24 then SGSN end) summarised_ontime_count, sum(entries) num_of_sumrows from NORTEL_PSCORE.GSCGMM_DY where
datetime = trunc(sysdate)-1 )sumt union all SELECT 'NORTEL_PSCORE.GSCSM_ACT' AS TABLENAME, sumday, CASE WHEN percent(summarised_ontime_count, (ontime_count-late_count), 2) > 100 THEN 100 ELSE
percent(summarised_ontime_count, (ontime_count-late_count), 2) END percent_of_cutoff, case when 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) < 0 then 0 else 100-percent(summarised_ontime_count,
(ontime_count-late_count), 2) end percent_of_cutoff_not_summed, percent(summarised_ontime_count, ontime_count+late_count, 2)percentage_of_total_by5am, percent(summarised_ontime_count+summarised_late_count,
ontime_count+late_count, 2) current_percentage, ontime_count+late_count current_raw_elements, summarised_ontime_count+summarised_late_count current_summary_elements, summarised_ontime_count-summarised_late_count
SUM_ELEMENTS_AT5AM, late_count late_raw_elements, ontime_count-late_count available_at_cutoff, percent(num_of_sumrows, num_of_rows, 2) accuracy from ( select count(distinct case when trunc(datetime_ins,'mi') >
trunc(sysdate)+3/24 then SGSN end) late_count, count(distinct case when trunc(datetime_ins,'mi') < trunc(sysdate)+3/24 then SGSN end) ontime_count, count (*) num_of_rows from NORTEL_PSCORE.GSCSM_ACT where datetime
between trunc(sysdate)-1 and trunc(sysdate)-1/24/60 )rawt, ( select max(datetime)sumday, count(distinct case when trunc(datetime_ins,'mi') > trunc(sysdate)+7/24 then SGSN end) summarised_late_count, count(distinct case when
trunc(datetime_ins,'mi') < trunc(sysdate)+7/24 then SGSN end) summarised_ontime_count, sum(entries) num_of_sumrows from NORTEL_PSCORE.GSCSM_ACT_DY where datetime = trunc(sysdate)-1 )sumt union all SELECT
'NORTEL_PSCORE.GSCSM' AS TABLENAME, sumday, CASE WHEN percent(summarised_ontime_count, (ontime_count-late_count), 2) > 100 THEN 100 ELSE percent(summarised_ontime_count, (ontime_count-late_count), 2) END
percent_of_cutoff, case when 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) < 0 then 0 else 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end percent_of_cutoff_not_summed,
percent(summarised_ontime_count, ontime_count+late_count, 2)percentage_of_total_by5am, percent(summarised_ontime_count+summarised_late_count, ontime_count+late_count, 2) current_percentage, ontime_count+late_count
current_raw_elements, summarised_ontime_count+summarised_late_count current_summary_elements, summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM, late_count late_raw_elements, ontime_count-
late_count available_at_cutoff, percent(num_of_sumrows, num_of_rows, 2) accuracy from ( select count(distinct case when trunc(datetime_ins,'mi') > trunc(sysdate)+3/24 then SGSN end) late_count, count(distinct case when
trunc(datetime_ins,'mi') < trunc(sysdate)+3/24 then SGSN end) ontime_count, count (*) num_of_rows from NORTEL_PSCORE.GSCSM where datetime between trunc(sysdate)-1 and trunc(sysdate)-1/24/60 )rawt, ( select
max(datetime)sumday, count(distinct case when trunc(datetime_ins,'mi') > trunc(sysdate)+7/24 then SGSN end) summarised_late_count, count(distinct case when trunc(datetime_ins,'mi') < trunc(sysdate)+7/24 then SGSN end)
summarised_ontime_count, sum(entries) num_of_sumrows from NORTEL_PSCORE.GSCSM_DY where datetime = trunc(sysdate)-1 )sumt union all SELECT 'NORTEL_PSCORE.GSDSTATS' AS TABLENAME, sumday, CASE WHEN
percent(summarised_ontime_count, (ontime_count-late_count), 2) > 100 THEN 100 ELSE percent(summarised_ontime_count, (ontime_count-late_count), 2) END percent_of_cutoff, case when 100-percent(summarised_ontime_count,
(ontime_count-late_count), 2) < 0 then 0 else 100-percent(summarised_ontime_count, (ontime_count-late_count), 2) end percent_o f_cutoff_not_summed, percent(summarised_ontime_count, ontime_count+late_count,
2)percentage_of_total_by5am, percent(summarised_ontime_count+summarised_late_count, ontime_count+late_count, 2) current_percentage, ontime_count+late_count current_raw_elements,
summarised_ontime_count+summarised_late_count current_summary_elements, summarised_ontime_count-summarised_late_count SUM_ELEMENTS_AT5AM, late_count late_raw_elements, ontime_count-late_count available_at_cutoff,
percent(num_of_sumrows, num_of_rows, 2) accuracy from ( select count(distinct case when trunc(datetime_ins,'mi') > trunc(sysdate)+3/24 then SGSN end) late_count, count(distinct case when trunc(datetime_ins,'mi') <
trunc(sysdate)+3/24 then SGSN end) ontime_count, count (*) num_of_rows from NORTEL_PSCORE.GSDSTATS where datetime between trunc(sysdate)-1 and trunc(sysdate)-1/24/60 )rawt, ( select max(datetime)sumday, count(distinct
case when trunc(datetime_ins,'mi') > trunc(sysdate)+7/24 then SGSN end) summarised_late_count, count(distinct case when trunc(datetime_ins,'mi') < trunc(sysdate)+7/24 then SGSN end) summarised_ontime_count, sum(entries)
num_of_sumrows from NORTEL_PSCORE.GSDSTATS_DY where datetime = trunc(sysdate)-1 )sumt
```

The Bug's Qualifications

- Not discovered since version 9i: Probably older
- Not recognized before by any Oracle customer
- What it took to find the root cause
  - 11 weeks
  - One Oracle ACE Director: Dan Morgan
  - Four dedicated Oracle support personnel (two on-site full-time)
  - Seven Production DBAs (part time)
  - Two members of Oracle Development
  - Support from Oak Table Members
    - Arup Nanda
    - Jonathan Lewis
    - Riyaj Shamsudeen
    - Tanel Poder
  - Andy Mendelsohn's personal involvement

First Sighting

```
Thu Feb 03 19:31:42 2011
SUCCESS: diskgroup OPMFRA01 was dismounted
Thu Feb 03 19:52:20 2011
Trace dumping is performing id=[cdmp_20110217195220]
Thu Feb 03 19:56:10 2011
Errors in file /appl/oracle/diag/rdbms/opm01p/opm01p5/trace/opm01p5_ora_21370.trc  (incident=1729262):
ORA-00600: internal error code, arguments: [ORA-00600: internal error code, arguments: [2252], [2835], [3070900825], [], [], [], [], [], [], [], [], [], [], [], [], [], [], [], []
ORA-02063: preceding line from NDR_LINK
Incident details in: /appl/oracle/diag/rdbms/opm01p/opm01p5/incident/incdir_1729262/opm01p5_ora_21370_i1729262.trc
Thu Feb 03 20:01:00 2011
Thread 5 advanced to log sequence 25931 (LGWR switch)
  Current log# 507 seq# 25931 mem# 0: +OPMREDO_A/opm01p/onlinelog/group_507.262.734230547
  Current log# 507 seq# 25931 mem# 1: +OPMREDO_B/opm01p/onlinelog/group_507.262.734230897
Thu Feb 03 20:02:08 2011
SUCCESS: diskgroup OPMFRA01 was mounted
```

Initial Work

- Oracle quickly identified the source of the ORA-600
- SCN max rate before 11.2.0.2 was 16,384/sec.
  (983,040/min. - 58,982,400/hr. - 1,415,577,600/day)
  - As of 11.2.0.2 max rate increases to 32,768/sec.
  - We were inserting 9,000,000,000/day
- The issue has only been reported once before and that was in 8.1.5 by a company with only ~200 databases: We have more than 10K
- Database links distribute the maximum SCN
- The term "SCN Jump" is invented
- Senior management gets **very** involved
- Senior management refuses to allow "best practices"
- Weeks wasted tracing irrelevancies

Database Links Internals

Relevance of System Change Numbers for In-Doubt Transactions

A system change number (SCN) is an internal timestamp for a committed version of the database. The Oracle Database server uses the SCN clock value to guarantee transaction consistency. For example, when a user commits a transaction, the database records an SCN for this commit in the redo log.

The database uses SCNs to coordinate distributed transactions among different databases. For example, the database uses SCNs in the following way:

* **An application establishes a connection using a database link.**
* **The distributed transaction commits with the highest global SCN among all the databases involved.**
* **The commit global SCN is sent to all databases involved in the transaction.**

SCNs are important for distributed transactions because they function as a synchronized commit timestamp of a transaction, even if the transaction fails.

- Note: This does not apply for JDBC, ODBC, or other connection methods

SCN Tracker

```
BEGIN
  FOR i IN 1..2 LOOP
    sd0 := SYSDATE;
    IF i = 1 THEN
      sd1 := sd0;
    ELSE
      sd2 := sd0;
    END IF;|
    cur_yr := TO_NUMBER(TO_CHAR(sd0, 'YYYY'));
    cur_mo := TO_NUMBER(TO_CHAR(sd0, 'MM'));
    cur_dy := TO_NUMBER(TO_CHAR(sd0, 'DD'));
    cur_hr := TO_NUMBER(TO_CHAR(sd0, 'HH24'));
    cur_mi := TO_NUMBER(TO_CHAR(sd0, 'MI'));
    cur_ss := TO_NUMBER(TO_CHAR(sd0, 'SS'));
    cur_scn := dbms_flashback.get_system_change_number;
    max_scn :=
      (((((((cur_yr-1988)*12*31)+((31*(cur_mo-1))+(cur_dy-1)))*24*60*60)+((((cur_hr*60)+cur_mi)*60)+cur_ss))*1024*16));
    IF i = 1 THEN
      cs1 := cur_scn;
      ms1 := max_scn;
      headrm1 := max_scn - cur_scn;
    ELSE
      cs2 := cur_scn;
      ms2 := max_scn;
      headrm2 := max_scn - cur_scn;
    END IF;
    IF i = 1 THEN
      dbms_lock.sleep(5*60); -- wait 5 minutes and run again
    END IF;
  END LOOP;
  IF headrm2 < headrm1 THEN
    burnrate := (headrm1-headrm2)/5;
    h2z := ROUND((headrm2/burnrate/1440),2)*24;
  ELSE
    burnrate := (headrm2-headrm1)/5;
    h2z := NULL;
  END IF;
  -- insert into logging table
```

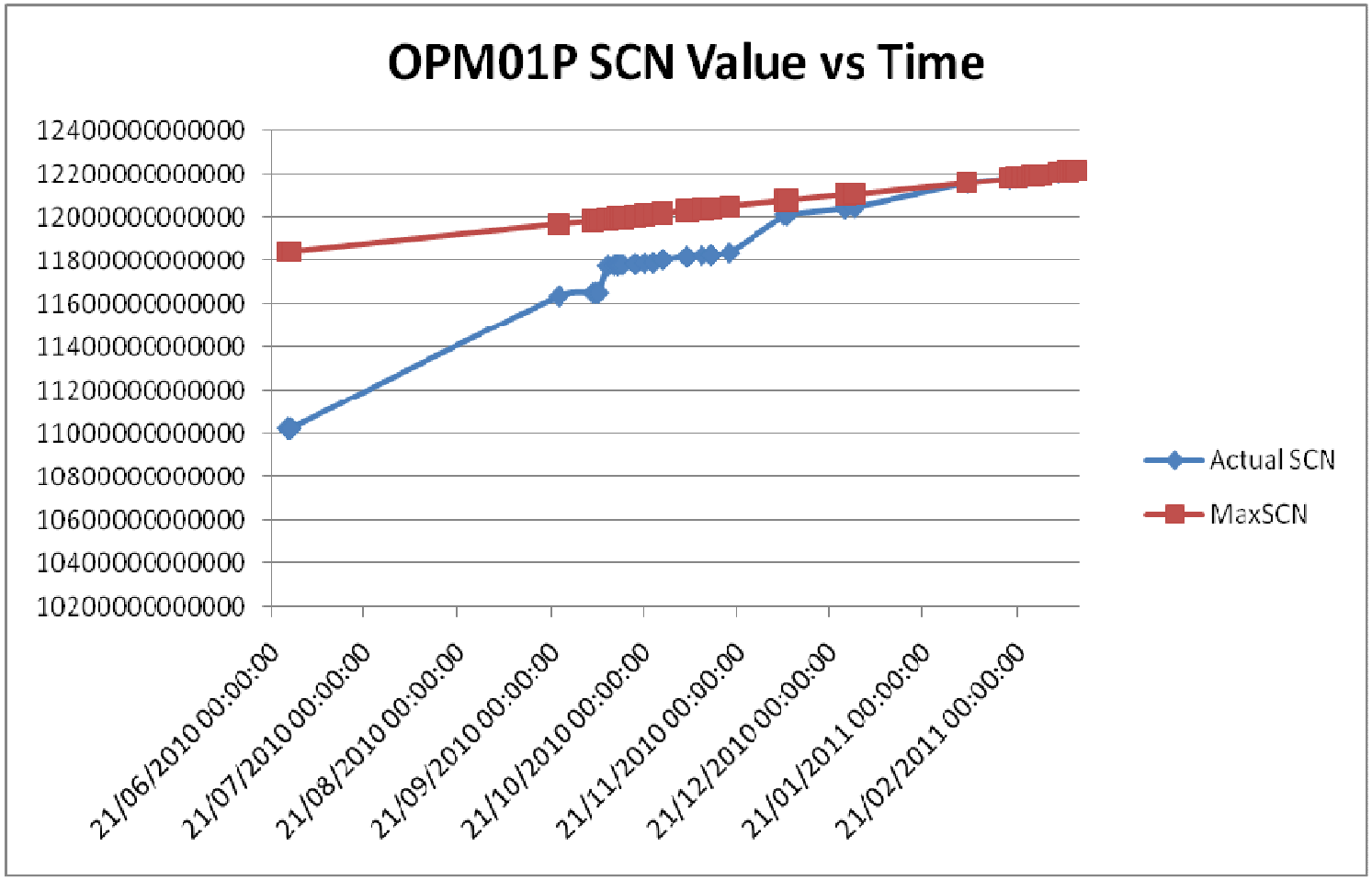
SCN Gap Tracking: Bad

```
SQL> select * from v_scn_gap
  2  where test2_datetime > sysdate-1;

TEST2_DATETIME       TEST2_HEADROOM RATE_PER_MIN RATE_PER_HR HOURS2ZERO
-------------------- -------------- ------------ ----------- ----------
26-APR-2011 17:35:00      682913527     521989.8    31319388      21.84
26-APR-2011 17:50:00      675150791    1228854.4    73731264       9.12
26-APR-2011 18:05:00      666553560       591649    35498940      18.72
26-APR-2011 18:20:00      662286655     938904.6    56334276      11.76
26-APR-2011 18:35:00      652174145     470405.0    28224300      23.04
26-APR-2011 18:50:00      647104628     652559.0    39153540      16.56
26-APR-2011 19:05:00      636368986     274281.6    16456896      38.64
26-APR-2011 19:20:00      639304415      44518.4     2671104
26-APR-2011 19:50:08      634797427     450686.4    27041184
26-APR-2011 20:05:54      630221617     408889.2    24533352
26-APR-2011 20:20:00      641108863     750031.0    45001860
26-APR-2011 20:35:00      652313458     810217.0    48613020
26-APR-2011 20:50:00      662684783     433170.4    25990224
26-APR-2011 21:05:06      674229682     938459.4    56307564


TEST2_DATETIME       TEST2_HEADROOM RATE_PER_MIN RATE_PER_HR HOURS2ZERO
-------------------- -------------- ------------ ----------- ----------
28-APR-2011 15:05:00     1881677494     833949.2    50036952
28-APR-2011 15:20:00     1893065104     753053.8    45183228
28-APR-2011 15:35:00     1904495482     780686.2    46841172
28-APR-2011 15:50:00     1917222386     843447.6    50606856
28-APR-2011 16:05:00     1930181269     848048.8    50882928
28-APR-2011 16:20:00     1942150863     715995.4    42959724
28-APR-2011 16:35:00     1953175650     842217.6    50533056
28-APR-2011 16:50:00     1950983064    1413509.0    84810540      23.04
28-APR-2011 17:05:00     1938373229     666124.4    39967464      48.48
28-APR-2011 17:20:00     1931490339     426728.2    25603692      75.36
28-APR-2011 17:35:00     1914255010     950190.8    57011448       33.6
28-APR-2011 17:50:00     1903398051     456508.0    27390480       69.6
28-APR-2011 18:05:00     1889420758     960524.4    57631464      32.88
28-APR-2011 18:20:00     1882512875     442962.0    26577720       70.8
```

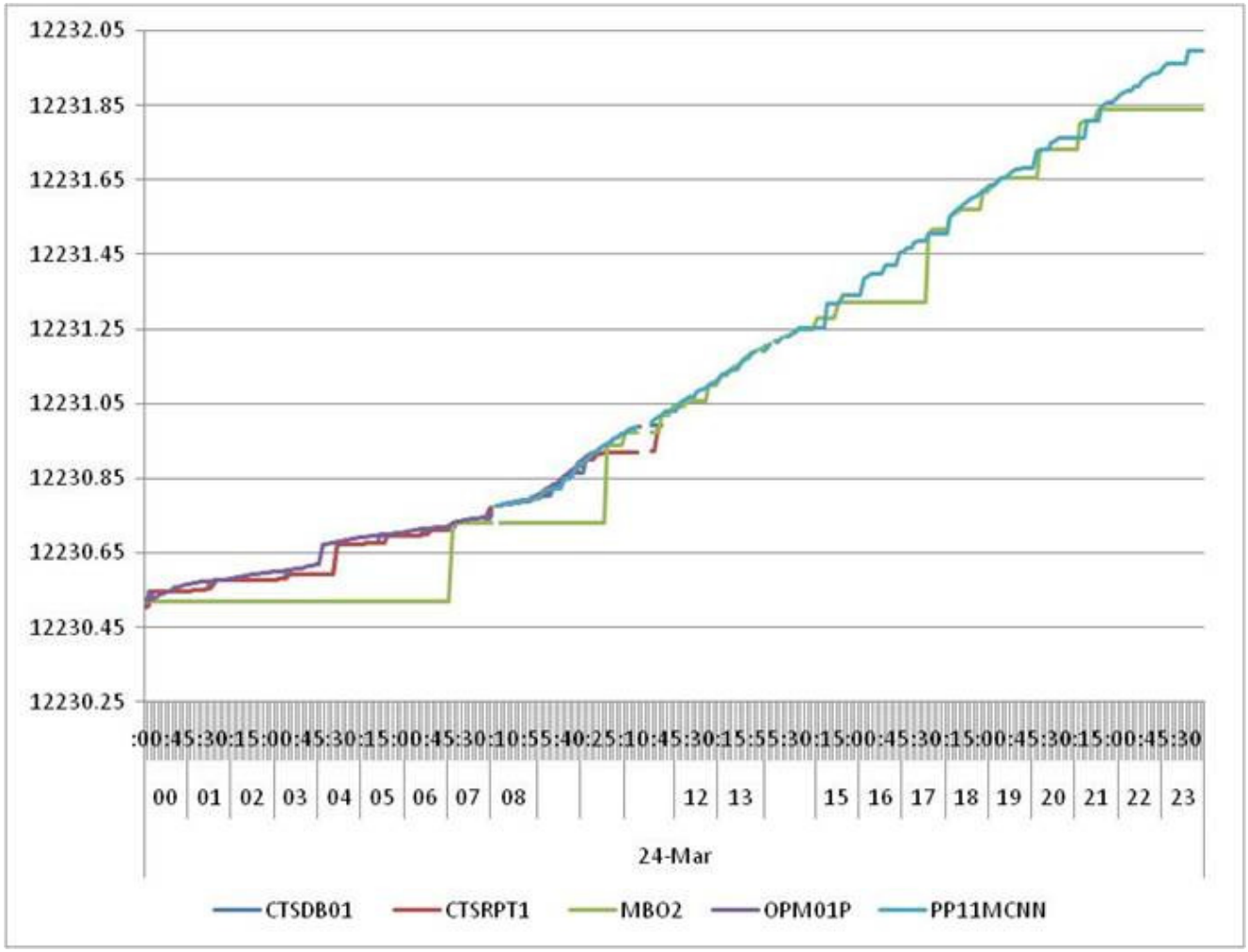
SCN Gap Tracking: Good

```
SQL> select * from v_scn_gap
  2  where test2_datetime > sysdate-1;

DATETIME                 HEADROOM RATE_PER_MIN RATE_PER_HR HOURS2ZERO
-------------------- ------------ ------------ ----------- ----------
11-MAY-2011 00:05:00   6506103188    6998200.6   419892036       15.6
11-MAY-2011 00:20:00   6513204788     543617.2    32617032
11-MAY-2011 00:35:00   6521420178     513123.6    30787416
11-MAY-2011 00:50:00   6530585913     717970.0    43078200
11-MAY-2011 01:05:00   6542639977     852059.4    51123564
11-MAY-2011 01:20:00   6541097520     863970.4    51838224
11-MAY-2011 01:35:00   6546037221     810748.6    48644916     134.64
11-MAY-2011 01:50:00   6559482107     896575.4    53794524
11-MAY-2011 02:05:00   6572030126     713590.6    42815436
11-MAY-2011 02:20:00   6584261177     849361.0    50961660
11-MAY-2011 02:35:00   6522396678     889341.8    53360508
11-MAY-2011 02:50:00   6535900852     897291.6    53837496
11-MAY-2011 03:05:00   6549401132     887350.6    53241036
11-MAY-2011 03:20:00   6562317808     768545.4    46112724
11-MAY-2011 03:35:00   6575539825     813914.6    48834876
11-MAY-2011 03:50:00   6589073166     885994.6    53159676
11-MAY-2011 04:05:00   6579304162    3760971.8   225658308       29.04
11-MAY-2011 04:20:00   6592771147     878893.6    52733616
11-MAY-2011 04:35:00   6606349369     901199.6    54071976
11-MAY-2011 04:50:00   6618097464     898662.0     5391972
11-MAY-2011 05:05:00   6631546687     882891.2    52973472
11-MAY-2011 05:20:00   6644969172     901559.4    54093564
11-MAY-2011 05:35:00   6636954732     827748.8    49664928
11-MAY-2011 05:50:00   6646938342     871115.2    52266912
11-MAY-2011 06:05:00   6642330448    2756499.2   165389952       40.08
11-MAY-2011 06:20:00   6654992990     806344.8    48380688
11-MAY-2011 06:35:00   6667497550     775057.0    46503420
11-MAY-2011 06:50:00   6680833195     907972.8    54478368
11-MAY-2011 07:05:00   6693534286     840129.0    50407740
11-MAY-2011 07:20:00   6694468241     651703.0    39102180
11-MAY-2011 07:35:00   6705466149     757110.8    45426648
11-MAY-2011 07:50:00   6717601759     801280.4     4807682
11-MAY-2011 08:05:00   6730036196     851263.6    51075816
11-MAY-2011 08:20:00   6742469369     810343.4    48620604
```

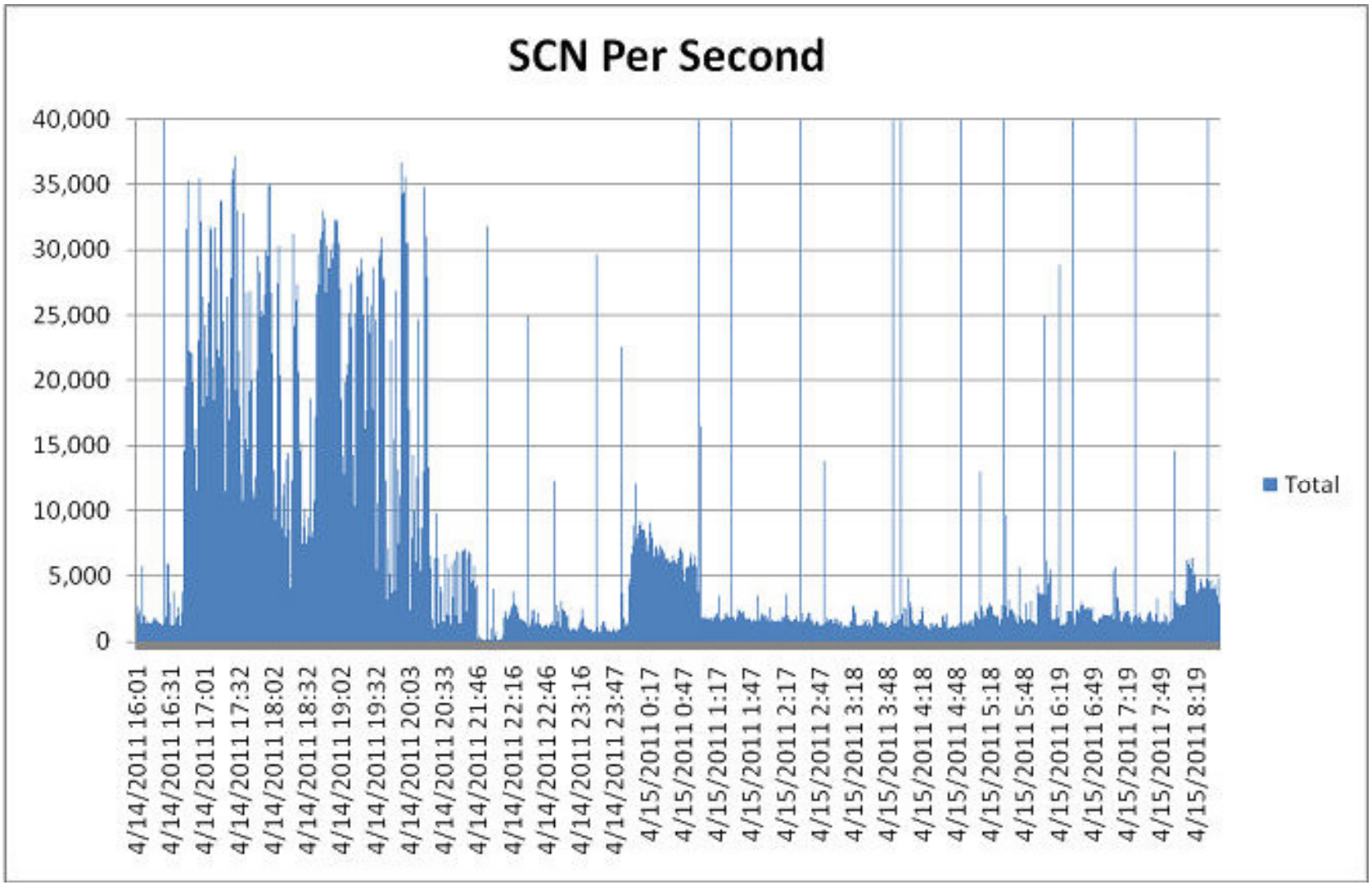
Actual vs. Max



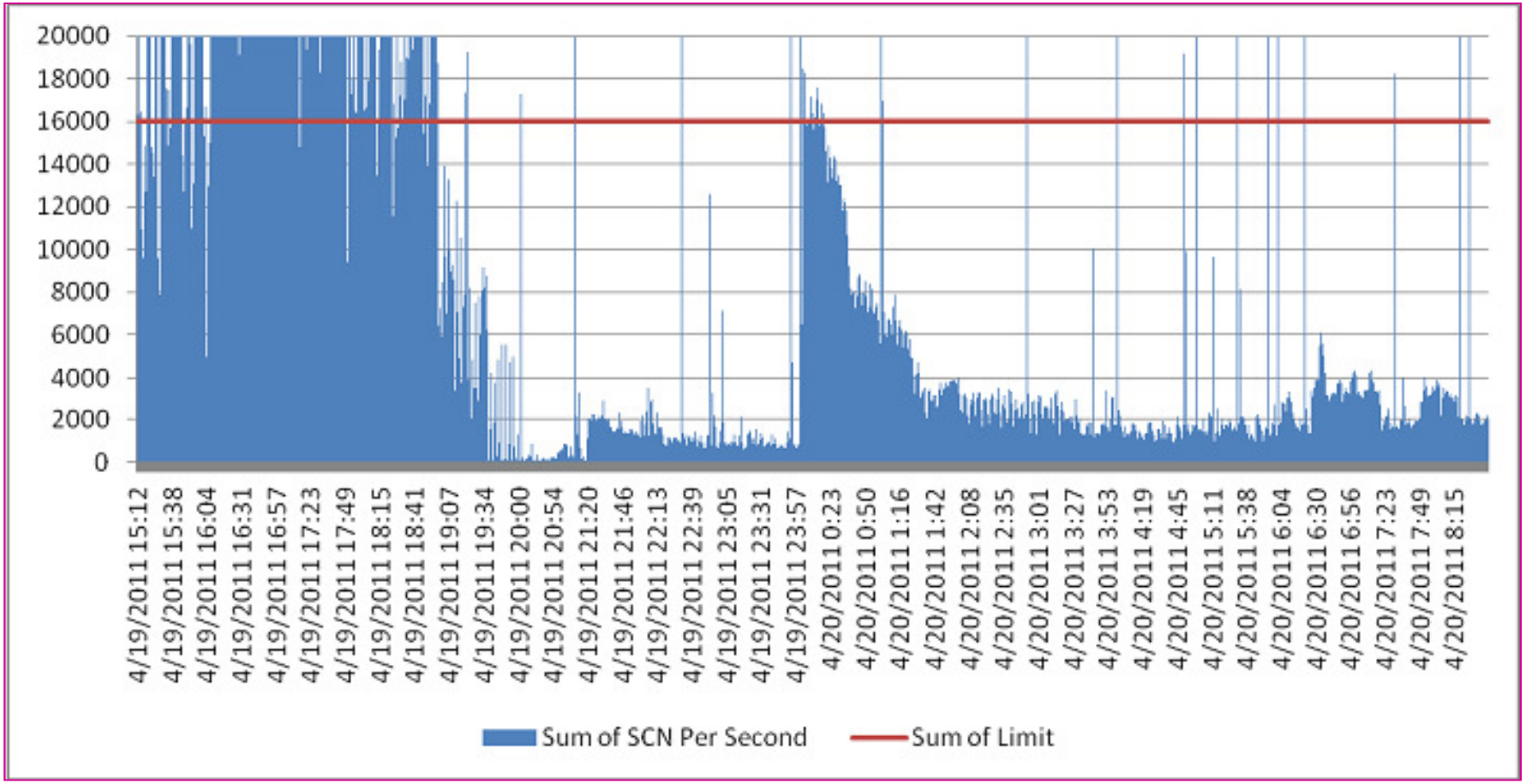
SCN Jump Visualized



Rolling Reboots



Maximum Execution Test



Oracle Support's Observations

From: SCOTT CARSTENS [mailto:scott.carstens@oracle.com]
Sent: Monday, April 18, 2011 6:48 AM
To: SHAMSUDEEN, RIYAJ Dan Morgan;
Cc:
Subject: Re: Few more debugging venues

Hi all,

I wanted to follow up on the discussion that we had on Friday regarding which sessions, post backup trigger, see increased scn generation and by how much. I'm attaching the analysis that was performed for the March 29th change, and as I mentioned last week a couple things really stick out about this, and we need to know if this behavior is consistent or not.

1. While overall scn utilization went up 4-6x (from 5k/sec to 20-30k/sec, the big scn consumers identified by Riyaj went up much more than this, around 100x for the [redacted] and [redacted] while the other processes [redacted] had very little prior history or appear to have started after the backup trigger had happened.

2. If you look at the graphs for these 4 processes, and summarize up the scn totals they generate, the 4 listed processes are responsible for 10-15k/sec scn rate. The 1 million level listed is for a 5 minute interval (if I read the data correctly) and you see consistent rates (with some spikes and some valleys) of anywhere from 3-5 million per period.

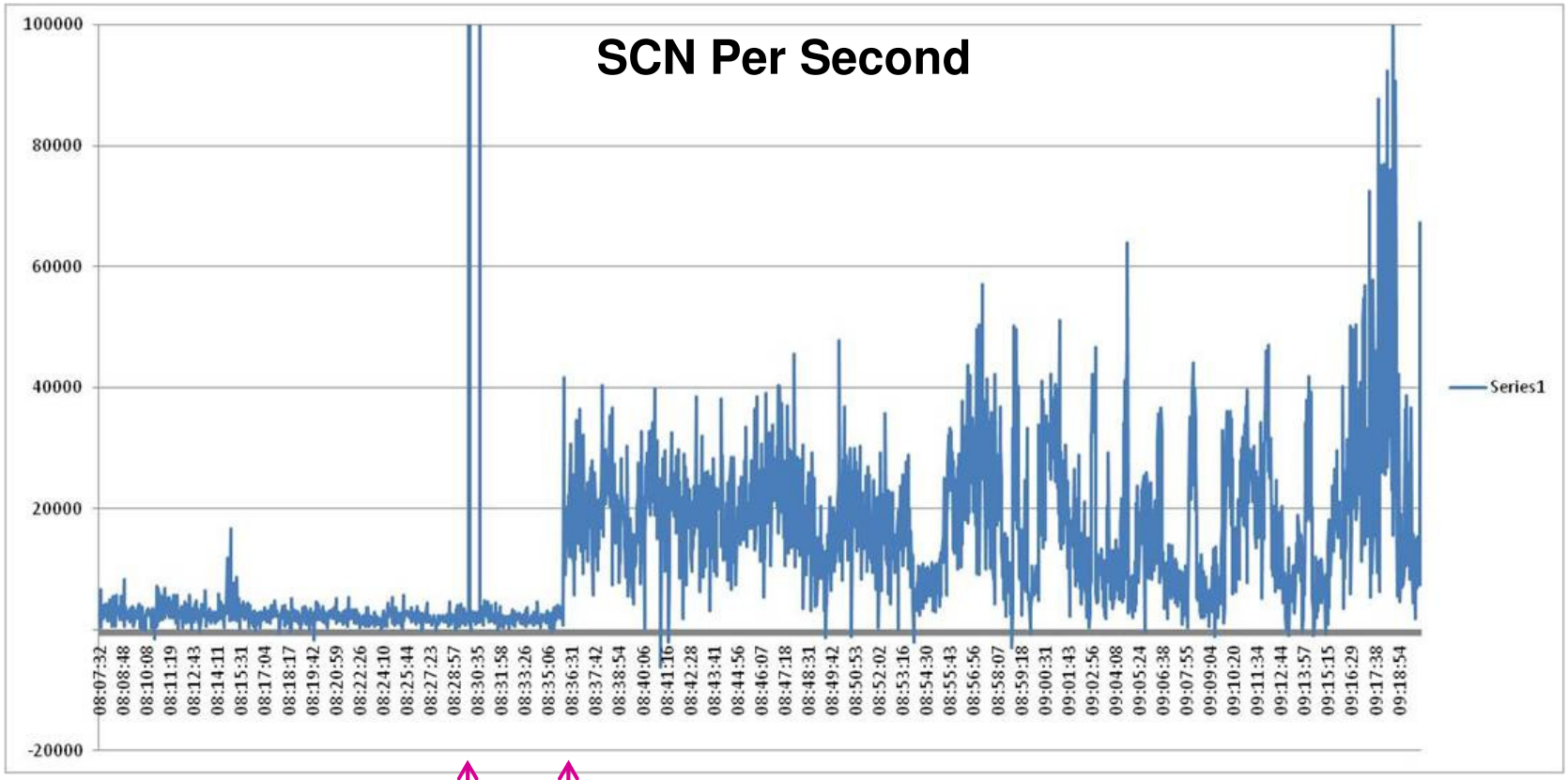
3. So at least in this example, you have:

1. 4 processes which appear responsible for the lion's share of the overall scn rate growth on the system.
2. Their individual increases were much greater than the overall system increase (multiple)

My understanding is that this was a consistent condition when the trigger occurs, but don't have any data to collaborate. We have this information for another couple backup triggers, and I believe it would be beneficial to get data that either confirms or refutes this. If we see that all the sessions are seeing their scn rates go up by around 4-5x, that is a different behavior to seeing a handful going up 100x.

Since we are also trying to identify targets to trace, if any pattern can be observed in what processes see increases is going to be helpful.

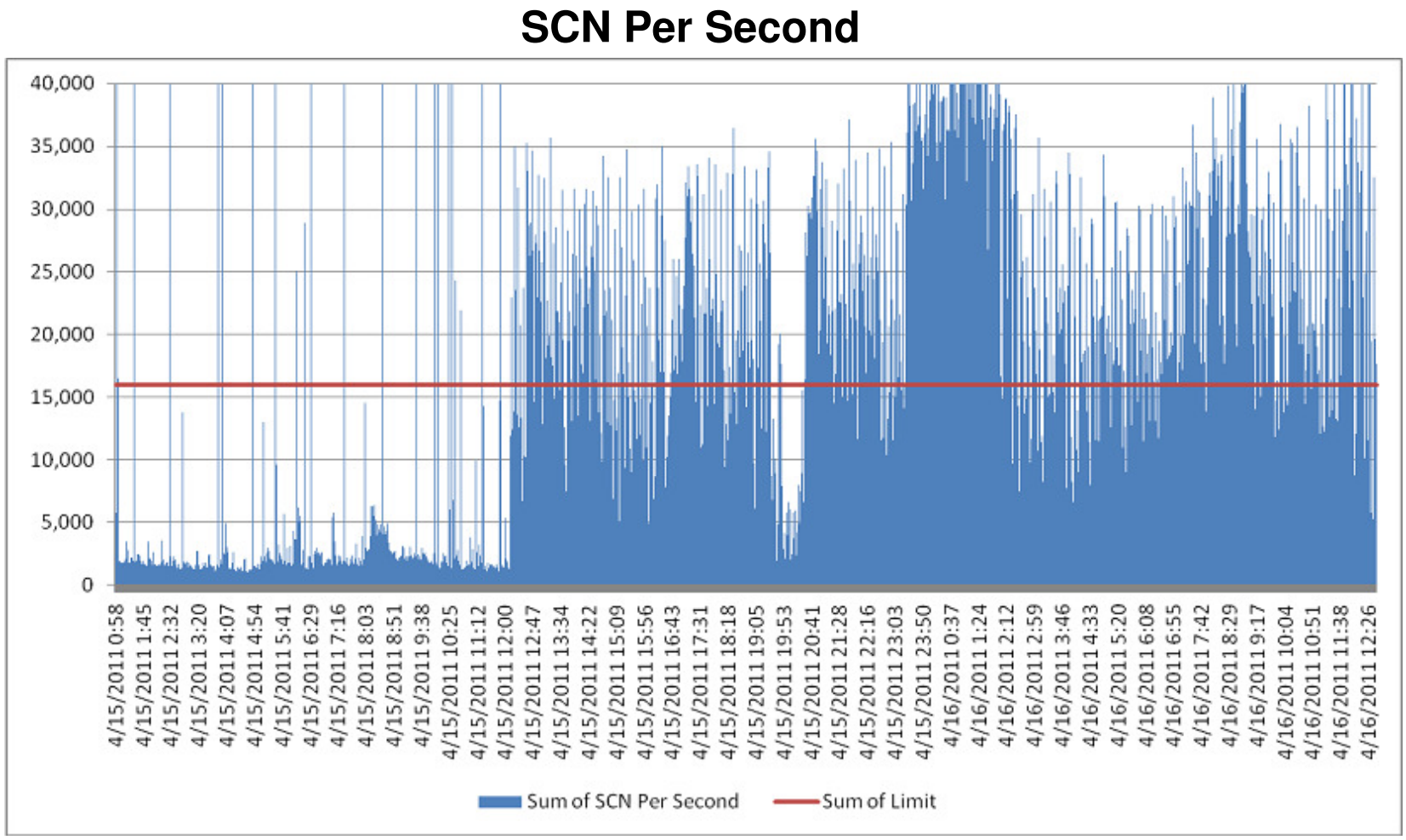
Smoking Gun 1



begin backup

end backup

Smoking Gun 2



Bug Description

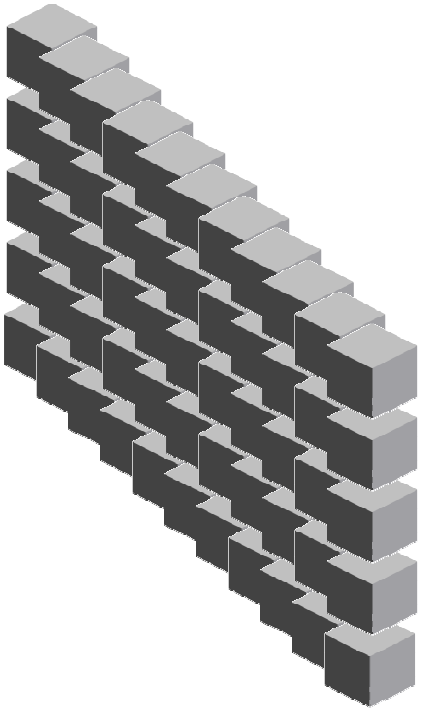
- Different in RAC and non-RAC
- A "feature" of all supported product versions
- When the database is put into "hot" backup mode SCN's are eaten at a greatly accelerated rate
- Only stopped by an instance reboot
- The enterprise impact can be made substantially worse in an environment with database links

Set and (oops) Forget

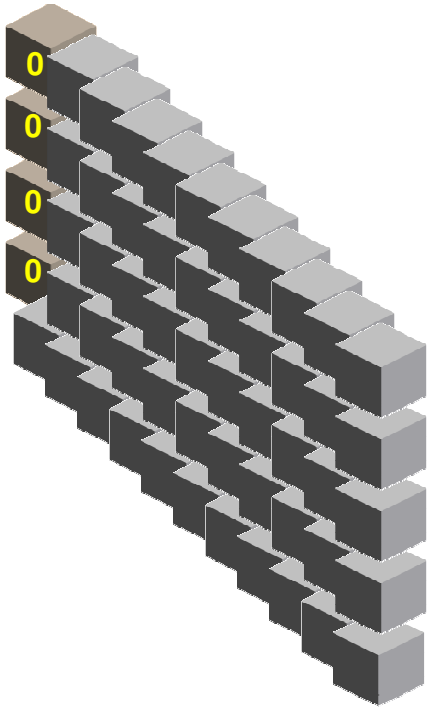
- **ALTER DATABASE BEGIN BACKUP;**
  - Set the flags to lock all datafiles from ID=1 to MAXDATAFILES
- Take Backup or Snap (and clone)
- **ALTER DATABASE END BACKUP;**
  - Unset the flag for all datafiles listed in file$

```
Tue Jan 18 18:20:18 2011
Create controlfile reuse set database "orabase"
MAXINSTANCES 8
MAXLOGHISTORY 1
MAXLOGFILES 16
MAXLOGMEMBERS 3
MAXDATAFILES 100
Datafile
'C:\oracle\oradata\orabase\SYSTEM01.DBF',
'C:\oracle\oradata\orabase\SYSAUX01.DBF',
'C:\oracle\oradata\orabase\UNDOTBS01.DBF',
'C:\oracle\oradata\orabase\USERS01.DBF'
LOGFILE GROUP 1 ('C:\oracle\oradata\orabase\redo01.log') SIZE 51200K,
GROUP 2 ('C:\oracle\oradata\orabase\redo02.log') SIZE 51200K,
GROUP 3 ('C:\oracle\oradata\orabase\redo03.log') SIZE 51200K RESETLOGS



SQL> select max(file#) from file$;

MAX(FILE#)
----------
         6
```

Lock 100 Datafiles



Unlock 4 Datafiles



Pre Patch

Post Patch

Patch# 12417252

```
SQL> oradebug setmypid
Statement processed.

SQL> oradebug dumpvar sga kcfdpk 10
kfil kcfdpk_ [380019860, 380019864) = 000000C8

SQL> oradebug dumpvar sga kcvblg 800
ub4 * kcvblg_ [38001AE30, 38001AE38) = 00000003 94599E58

SQL> alter database begin backup;

Database altered.

SQL> oradebug dumpvar sga kcfdpk 10
kfil kcfdpk_ [380019860, 380019864) = 000000C8

SQL> oradebug dumpvar sga kcvblg 800
ub4 * kcvblg_ [38001AE30, 38001AE38) = 00000003 94599E58

SQL> alter database end  backup;
Database altered.

SQL> oradebug dumpvar sga kcfdpk 10
kfil kcfdpk_ [380019860, 380019864) = 000000C8

SQL> oradebug dumpvar sga kcvblg 800
ub4 * kcvblg_ [38001AE30, 38001AE38) = 00000003 94599E58
```

```
SQL> oradebug setmypid
Statement processed.

SQL> oradebug dumpvar sga kcfdpk 10
kfil kcfdpk_ [380019860, 380019864) = 000000C8

SQL> oradebug dumpvar sga kcvblg 800
ub4 * kcvblg_ [38001AE30, 38001AE38) = 00000003 94581308

SQL> alter database begin backup;

Database altered.

SQL> oradebug dumpvar sga kcfdpk 10
kfil kcfdpk_ [380019860, 380019864) = 000000C8

SQL> oradebug dumpvar sga kcvblg 800
ub4 * kcvblg_ [38001AE30, 38001AE38) = 00000003 94581308

SQL> alter database end  backup;
Database altered.

SQL> oradebug dumpvar sga kcfdpk 10
kfil kcfdpk_ [380019860, 380019864) = 000000C8

SQL> oradebug dumpvar sga kcvblg 800
ub4 * kcvblg_ [38001AE30, 38001AE38) = 00000003 94581308
```

Questions

ERROR at line 1:

ORA-00028: your session has been killed

All demos at morganslibrary.org

- **Library**
- **How Can I?**

Thank you