Advanced Customer Services

Report of Findings For T24LIVE

Prepared For

Prepared by Advanced Customer Services

Author:	Do Duc Thinh
Creation Date:	24 May 2016
Version:	1.0
Data Collection:	From 28-APR-16 01:00 to 30-APR-16 23:30

1. Contents

2. Executive Summary

Oracle Advanced Customer Services (ACS) conducted a technical assessment on the Techcombank T24LIVE database hosted on AIX hosts named t24db01, t24db02. There were no changes made to the assessed system. This document recommends changes and identifies specific areas that require investigation that is more detailed.

The Engineered Assessment Performance (EAP) is a remote service that examines a customer system to make high-level performance recommendations and identify critical areas requiring immediate attention.
The goal of the technical assessment of the T24LIVE environment, i.e., host, database, and I/O subsystem, was to identify factors that may be negatively affecting system performance and response time. Performance data was collected from 28-APR-16 01:00 to 30-APR-16 23:30.

Category	Information	Additional Comments
Database	Drop or change to UNUSABLE state with indexes have low distinct key	See 5.1
	Use bind variable instead of literal value	See Avoid using literal value on conditions with high distinct values
	Partition table with high concurrency access	See 5.4
	Optimize table FBNK_CATEG_ENT_ACTIVITY	See 5.2
	Optimize enq: HW - contention wait event	See 5.5
	Adjust sequence cache for SYS.IDGEN1$	See High SQ Enqueue Contention with LOBs

Other areas were also identified that can be examined by reviewing the Findings and Recommendations Section in detail.

3. System Reviewed

The following system was reviewed during this engagement:

3.1. Host

Server Name	t24db01, t24db02
Running DB Instances	1
Purpose (Production, development, Q&A)	Production
Platform	AIX-Based Systems (64-bit)
Model	
Operating System	AIX
O/S Version and Release	IBM AIX 6100-08-02-1316
Cluster Software / Version	Grid Infrastructure
# CPU	96
Processor / CPU Speed	PowerPC_POWER7 @ 3864 MHz
Memory	461 GB
Volume Manager / Version	

3.2. Database

Database Name	T24LIVE
Instance Name	t24live1, t24live2
RAC-Configuration	Yes
Machine Name	t24db01, t24db02
RDBMS Version/Release	11.2.0.4.0
Usage (OLTP, DSS, etc.)	OLTP
File System / raw devices	ASM
AIO	Yes
Disk Space (of all db files)	Datafiles: 3.6 TB Tempfiles: 38 GB
Archiving Enabled?	Yes

4. Overview

Unless otherwise noted, all findings are based on data collected from 28-APR-16 01:00 to 30-APR-16 23:30.

4.1. Database Response Time

Within the scope of the RDBMS, Response Time in its simplest form consists of Service Time + Wait Time. Service Time equates to time the request is actively being processed on the CPU, while Wait Time encompasses everything else. Oracle tracks Service Time in views related to system statistics and Wait Time within a set of views collectively known as the Wait Interface. By taking snapshots of the relevant views, deltas can be calculated and analyzed to explain precisely where user response time is being spent.

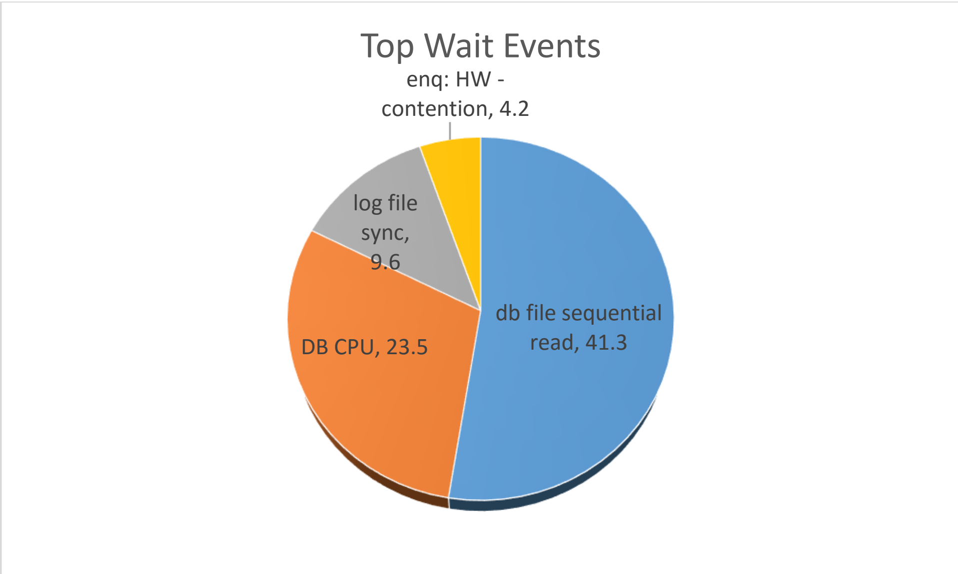

Figure 1: Top Foreground Waits for T24LIVE1

Figure 2: Top Foreground Waits for T24LIVE2

4.2. Connection Balance

It is advisable for Oracle RAC databases to have connections load balanced to make optimum utilization of resources. Given below is connection load balance graph for both the instances.

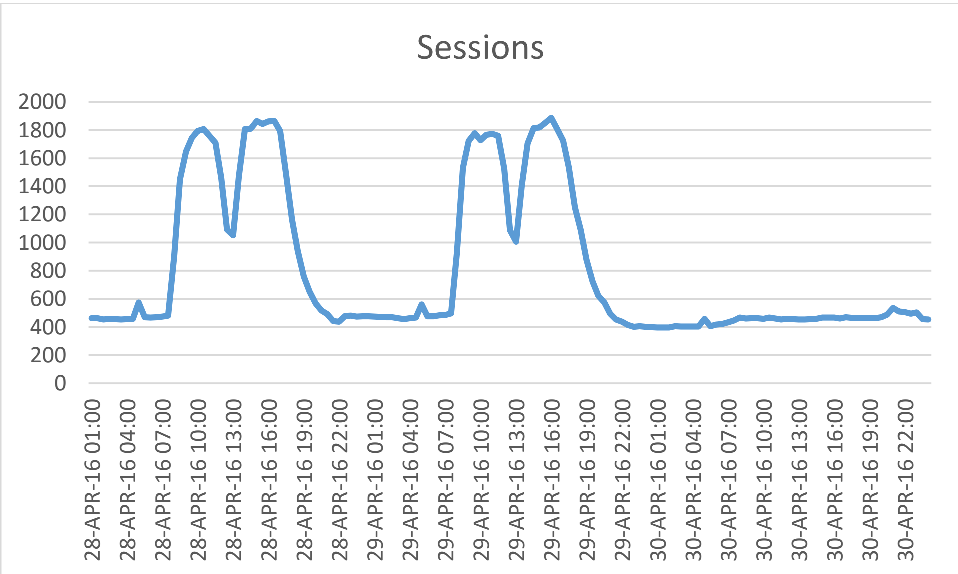

Figure 3: Average Database Sessions for T24LIVE1

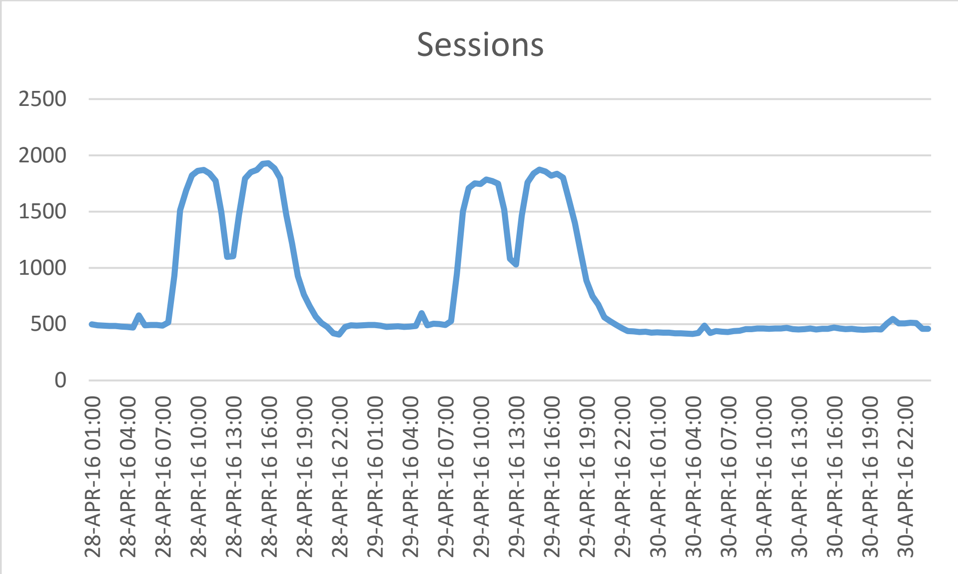

Figure 4: Average Database Sessions for T24LIVE2

Observation: As seen from the data, average number of session is evenly distributed on both the instances. This indicates that connections are successfully load balanced across all nodes in the cluster.

4.3. Average Active Session

The number of active session show how many users are waiting for Oracle to process it's task. The higher, the more load are put on database server.

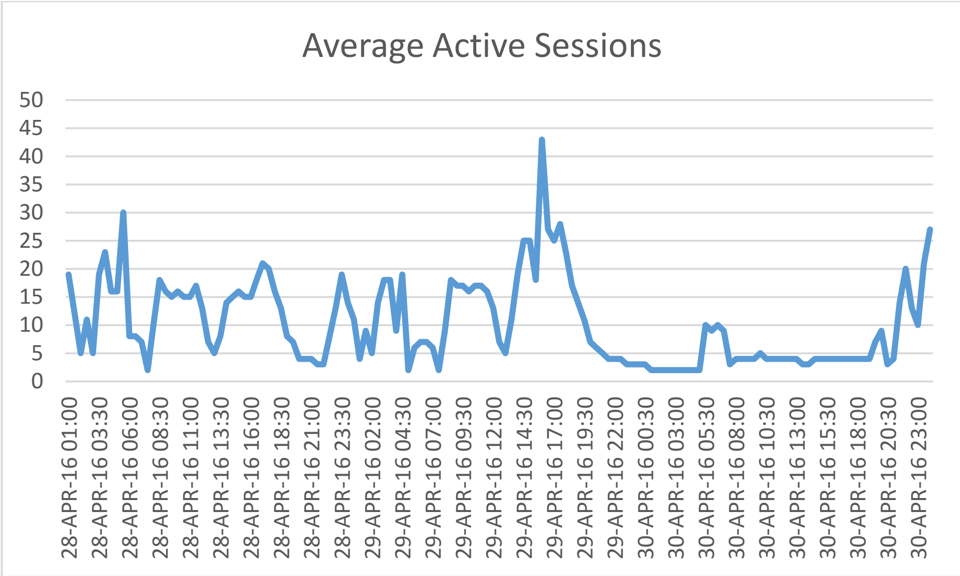

Figure 5: Average Active Sessions for T24LIVE1

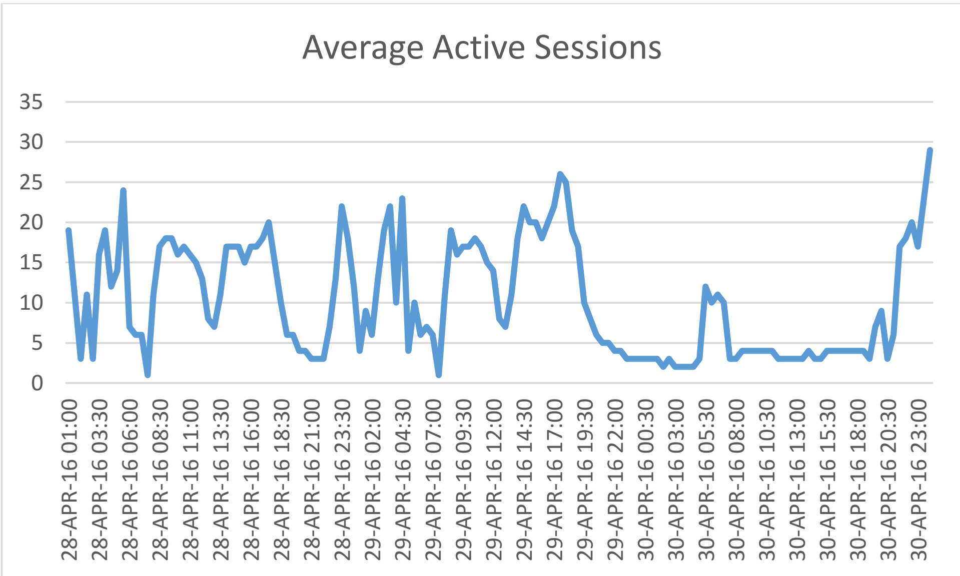

Figure 6: Average Active Sessions for T24LIVE2

Observation: Average active session is within system capacity

4.4. CPU Utilization

CPU capacity is a critical resource that should remain below a sustained rate of seventy percent at nearly all times. Whenever CPU utilization is over this amount, response time and throughput suffer, particularly if the saturation is sustained. This system was at all times well below seventy percent.

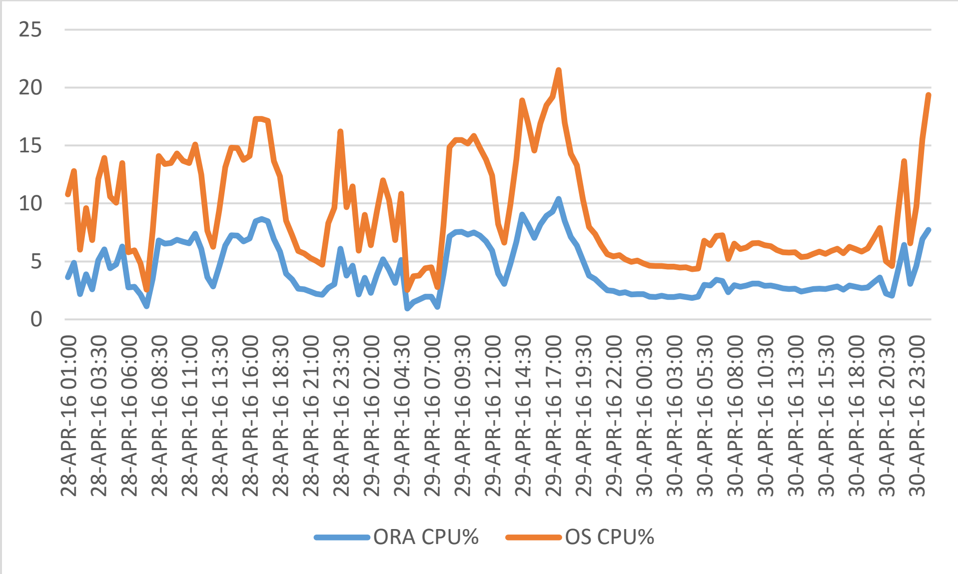

Figure 7: CPU Break Down for T24LIVE1

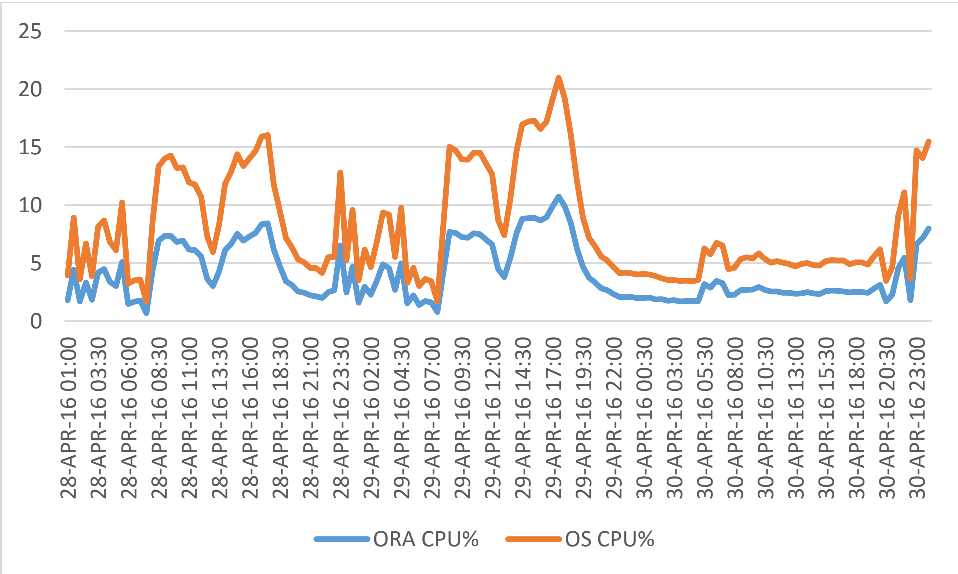

Figure 8: CPU Break Down for T24LIVE2

Observation:

The CPU utilization is well under server capacity.

4.5. Disk IO Activity

Physical Disk IO involves the transfer of data to or from the physical hardware. If a disk is more than 60% busy over sustained periods of time, this can indicate overuse of that resource.

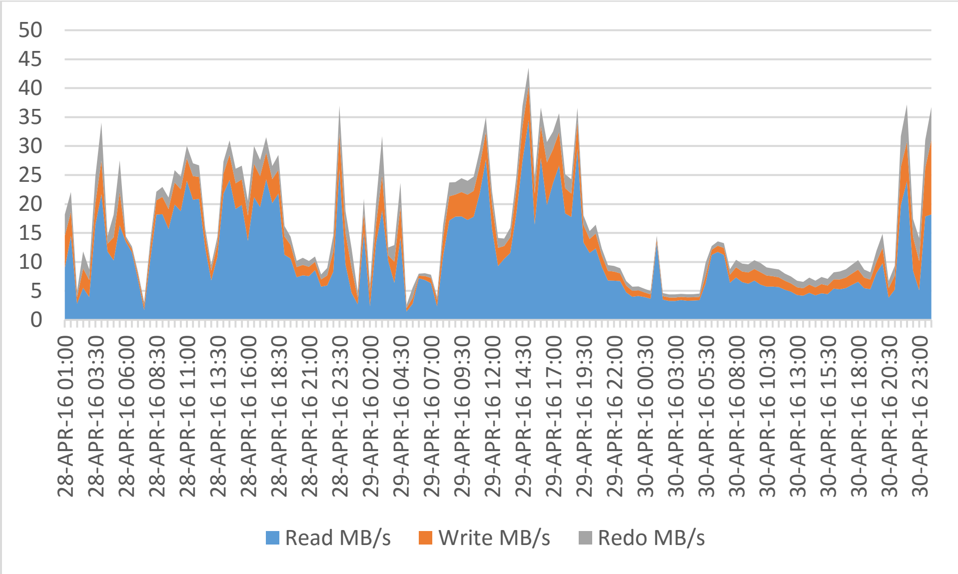

Figure 9: Disk IO for T24LIVE1

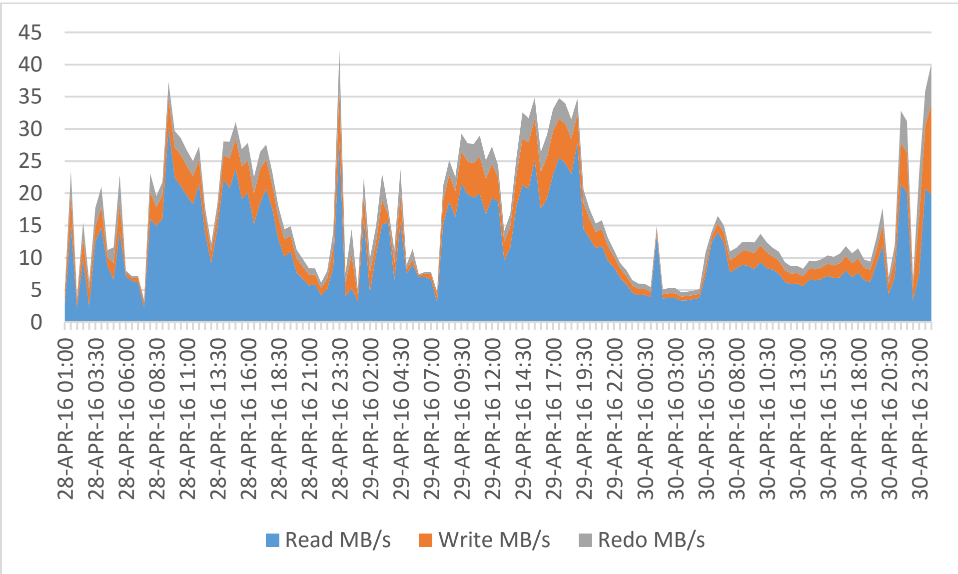

Figure 10: Disk IO for T24LIVE2

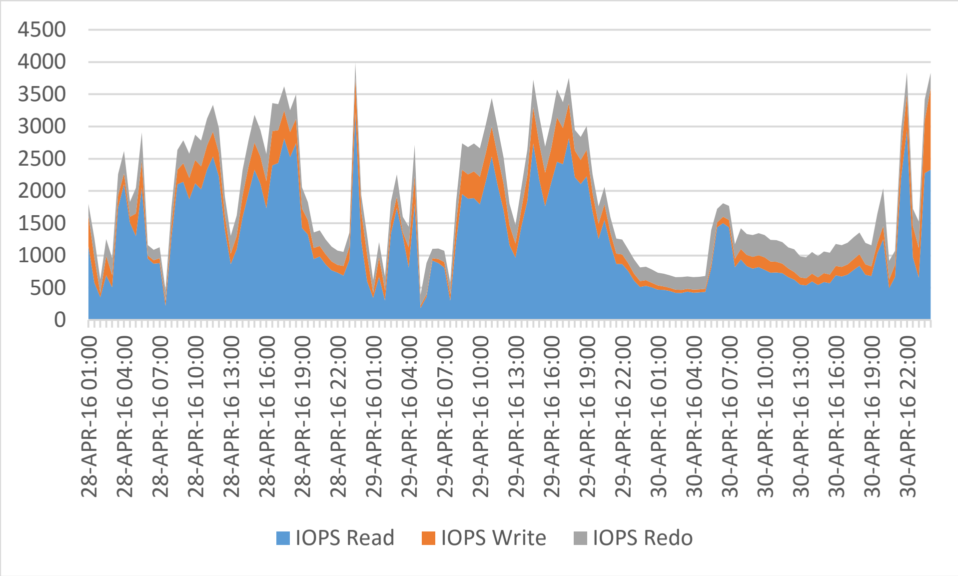

Figure 11: Read , Write Operations per Second (IOPS) for T24LIVE1

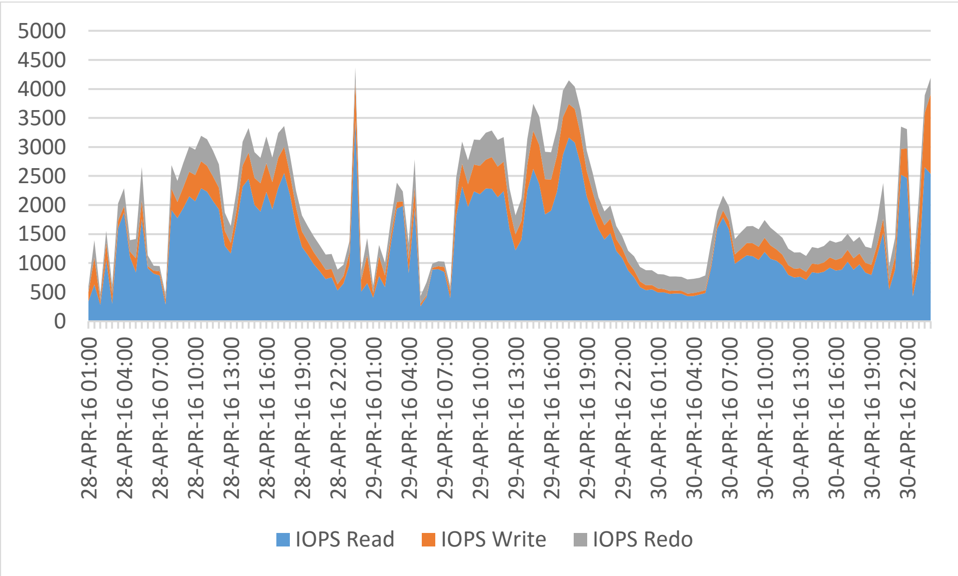

Figure 12: Read , Write Operations per Second (IOPS) for T24LIVE2

4.6. Logical IO

Oracle Logical I/O is defined as whenever the Oracle kernel requests access to an Oracle block in the database buffer cache. If the kernel cannot find a specified Oracle block in the database buffer cache, then the Logical I/O causes physical I/O. Because of this, Logical Reads is a better measurement of internal database activity than Physical Reads. Additionally, Logical Reads do require resources and affect response time to a much greater extent than once was thought.

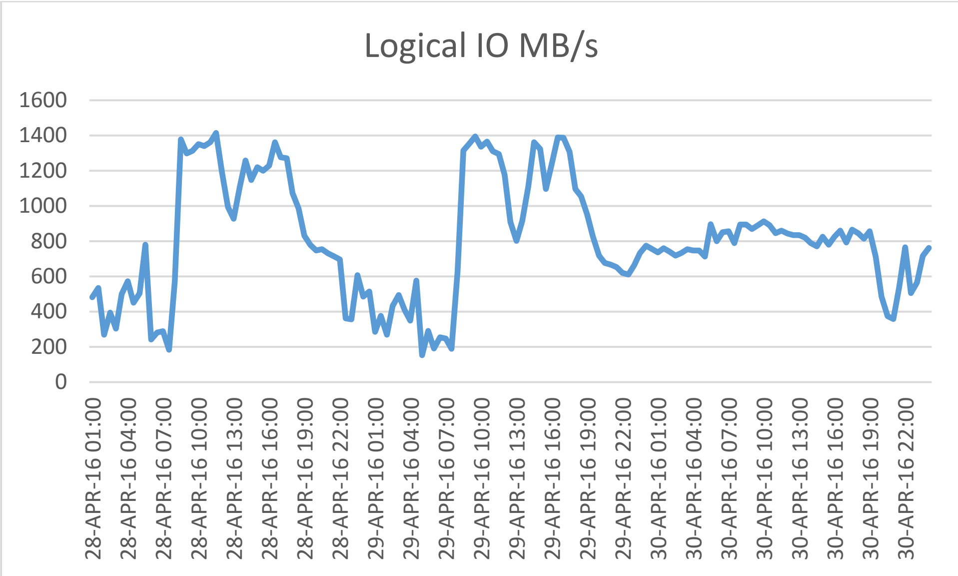

Figure 13: Logical IO for T24LIVE1

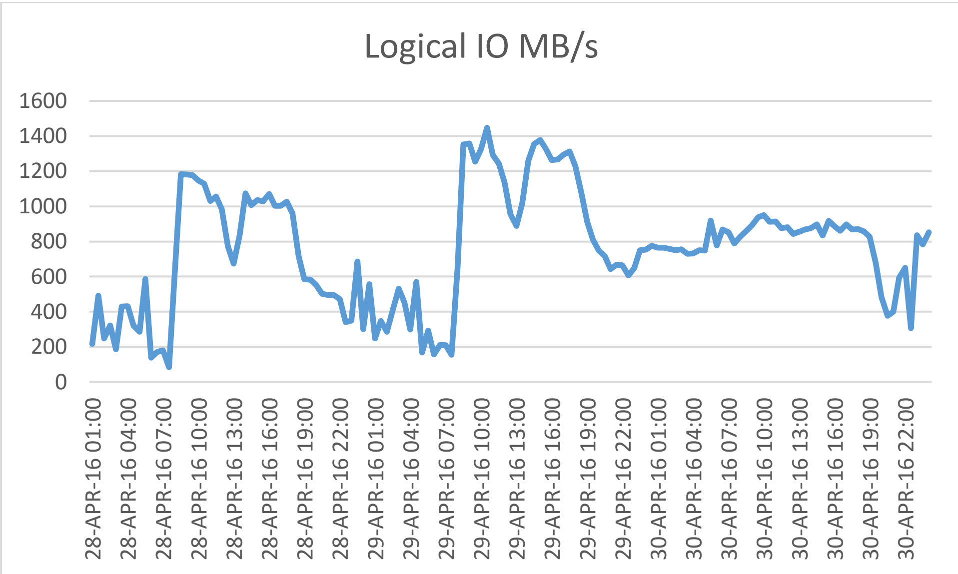

Figure 14: Logical IO for T24LIVE2

4.7. Transactions Rate and User Calls

Peak periods are often defined by the transaction arrival rate. In Oracle, a transaction is defined by a series of operations that result in either a COMMIT operation, or a ROLLBACK operation.

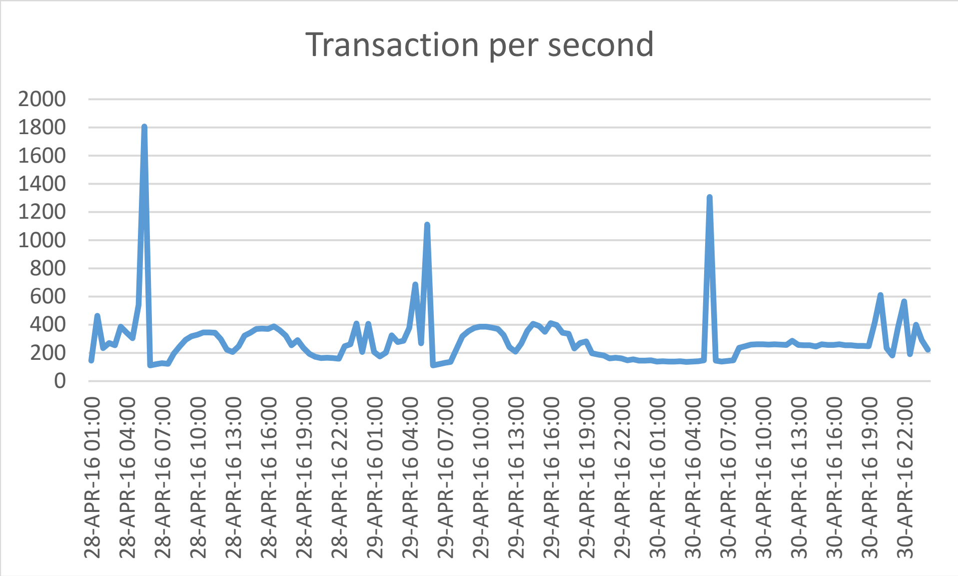

Figure 15: Transactions per Second for T24LIVE1

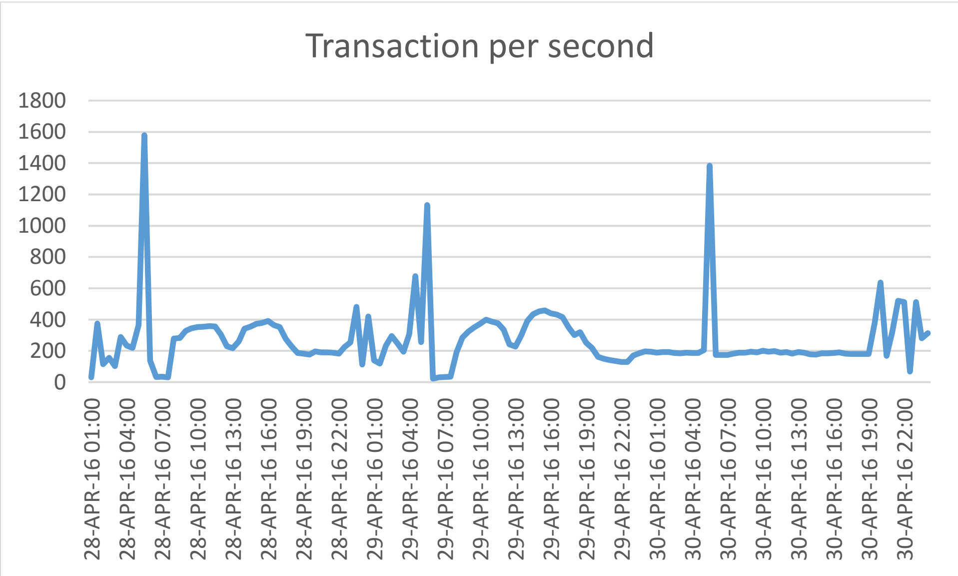

Figure 16: Transactions per Second for T24LIVE2

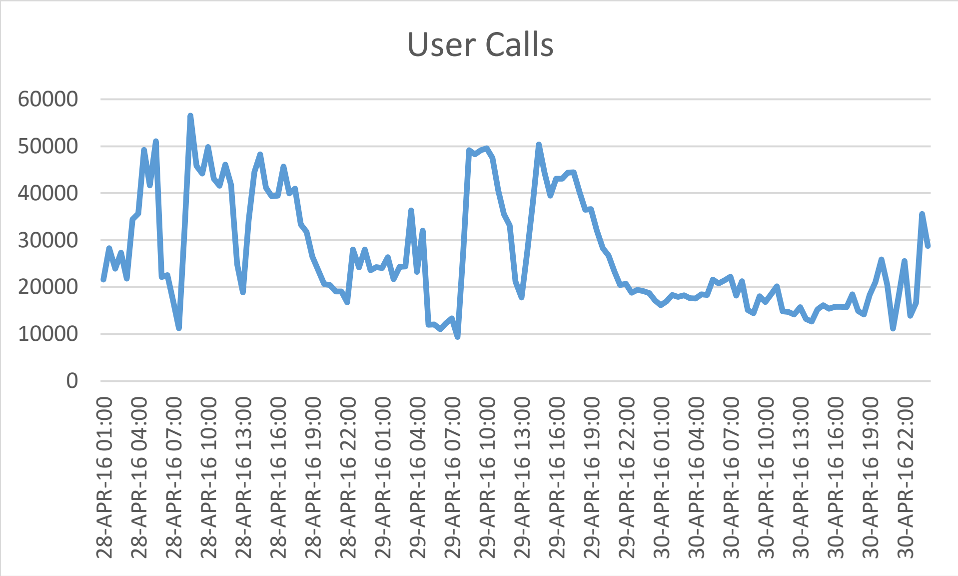

Figure 17: User call per seconds for T24LIVE1

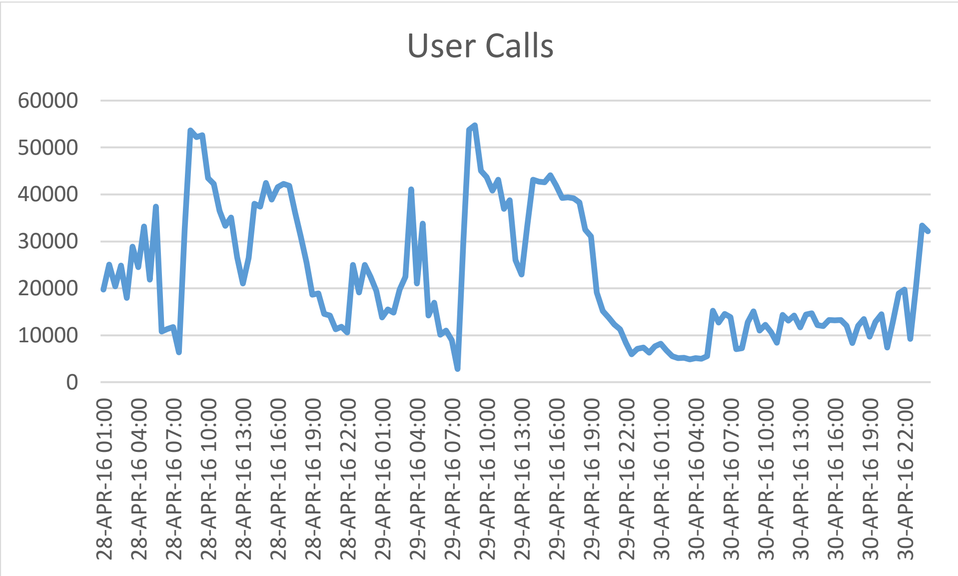

Figure 18: User call per seconds for T24LIVE2

4.8. PGA Memory Statistics

The Program Global Area (PGA) is a memory buffer that contains data and control information for a server process. A PGA is created by Oracle when a server process is started. The information in a PGA depends on the Oracle configuration.
Access to it is exclusive to that server process and is read from and written to only by the Oracle

code acting on behalf of it. An example of such information is the runtime area of a cursor. Each time a cursor is executed, a new runtime area is created for that cursor in the PGA memory region of the server process executing that cursor. Analyze this section helps when using the new model to allocate PGA.

The goal is to have most work areas running with an optimal size (for example, more than 90% or even 100% for pure OLTP systems), while a smaller fraction of them are running with a one-pass size (for example, less than 10%). Multi-pass execution should be avoided. Even for DSS systems running large sorts and hash-joins, the memory requirement for the one-pass executions is relatively small. A system configured with a reasonable amount of PGA memory should not need to perform multiple passes over the input data.

Under automatic PGA memory management mode, Oracle honors the PGA_AGGREGATE_TARGET limit by controlling dynamically the amount of PGA memory allotted to SQL database areas. At the same time, Oracle maximizes the performance of all the memory-intensive SQL operators by maximizing the number of database areas that are using an optimal amount of PGA memory (cache memory). The rest of the database areas are executed in one-pass mode, unless the PGA memory limit set by PGA_AGGREGATE_TARGET is so low that multipass execution is required to reduce even more the consumption of PGA memory and honor the PGA target limit.

In 11g, PGA_AGGREGATE_TARGET controls work areas allocated by both dedicated and shared connections.
This metric is computed by Oracle to reflect the performance of the PGA memory component. It is cumulative from instance start-up. A value of 100% means that all work areas executed by the system since instance start-up have used an optimal amount of PGA memory. This is, of course, ideal but rarely happens except maybe for pure OLTP systems. In reality, some work areas run one-pass or even multi-pass, depending on the overall size of the PGA memory. When a work area cannot run optimally, one or more extra passes is performed over the input data. This reduces the cache-hit percentage in proportion to the size of the input data and the number of extra passes performed.

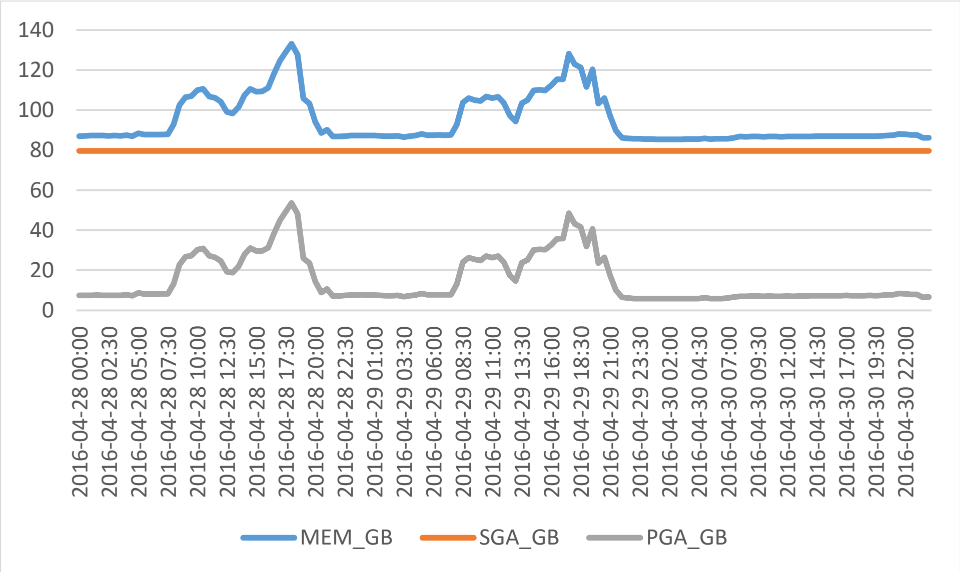

Figure 19: Memory allocated for T24LIVE1

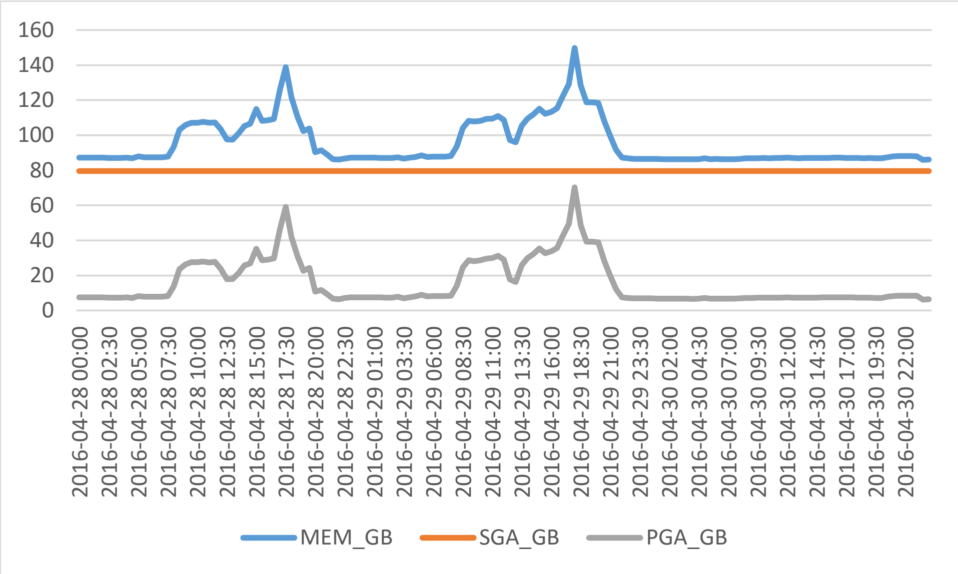

Figure 20: Memory allocated for T24LIVE2

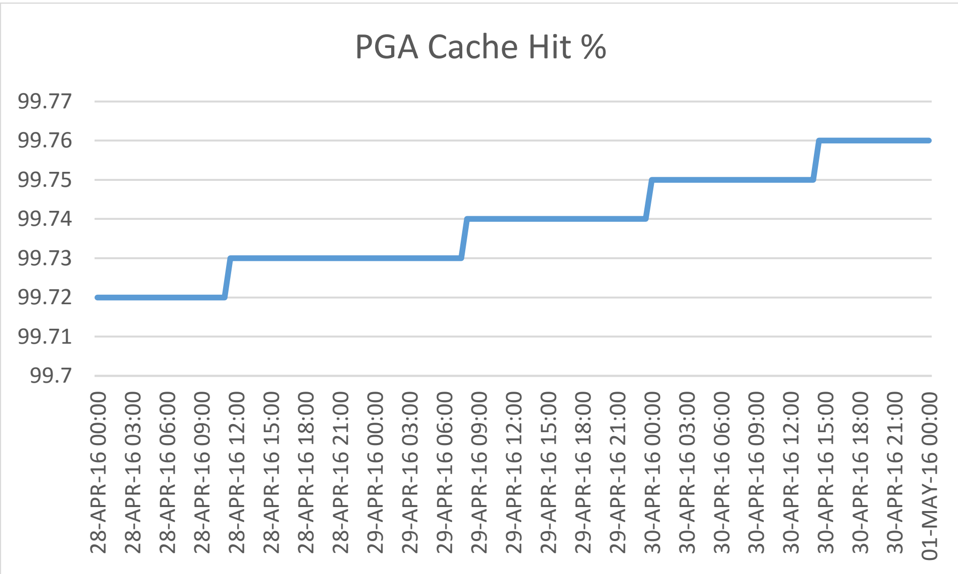

Figure 21: PGA cache hit % for T24LIVE1

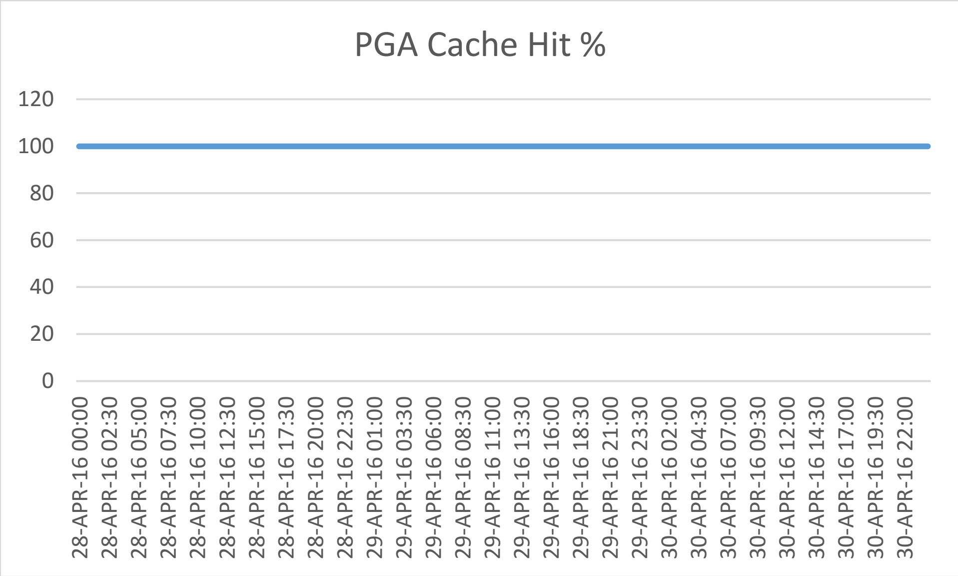

Figure 22: PGA cache hit % for T24LIVE2

Observation:

The current size of PGA is already optimal.

4.9. Redo Transaction Activity

Redo logs contain the transaction data that is created when users submit changes to the database. As each redo log fills to capacity with these changes, it "switches" to the next redo log to continue, while the just filled redo log is then copied by the archiving process. If redo log switching is occurring too frequently, slowdowns may be experienced while waiting on the archiver process to finish, or for redo log space manipulation to occur. Increasing or decreasing the size of the redo logs can easily adjust the rate of switching.

#	GROUP#	MEMBER	SIZE (GB)
1	15	+RECOT24R14_DC/t24r14dc/onlinelog/group_15.263.899392089	4
2	15	+T24R14_DC/t24r14dc/onlinelog/group_15.258.899392075	4
3	16	+RECOT24R14_DC/t24r14dc/onlinelog/group_16.262.899392113	4
4	16	+T24R14_DC/t24r14dc/onlinelog/group_16.259.899392101	4
5	17	+RECOT24R14_DC/t24r14dc/onlinelog/group_17.261.899392139	4
6	17	+T24R14_DC/t24r14dc/onlinelog/group_17.264.899392125	4
7	18	+RECOT24R14_DC/t24r14dc/onlinelog/group_18.260.899392163	4
8	18	+T24R14_DC/t24r14dc/onlinelog/group_18.261.899392151	4
9	25	+RECOT24R14_DC/t24r14dc/onlinelog/group_25.259.899392187	4
10	25	+T24R14_DC/t24r14dc/onlinelog/group_25.262.899392175	4
11	26	+RECOT24R14_DC/t24r14dc/onlinelog/group_26.258.899392211	4
12	26	+T24R14_DC/t24r14dc/onlinelog/group_26.265.899392199	4
13	27	+RECOT24R14_DC/t24r14dc/onlinelog/group_27.257.899392243	4

14	27	+T24R14_DC/t24r14dc/onlinelog/group_27.266.899392227	4
15	28	+RECOT24R14_DC/t24r14dc/onlinelog/group_28.256.899392269	4
16	28	+T24R14_DC/t24r14dc/onlinelog/group_28.263.899392257	4

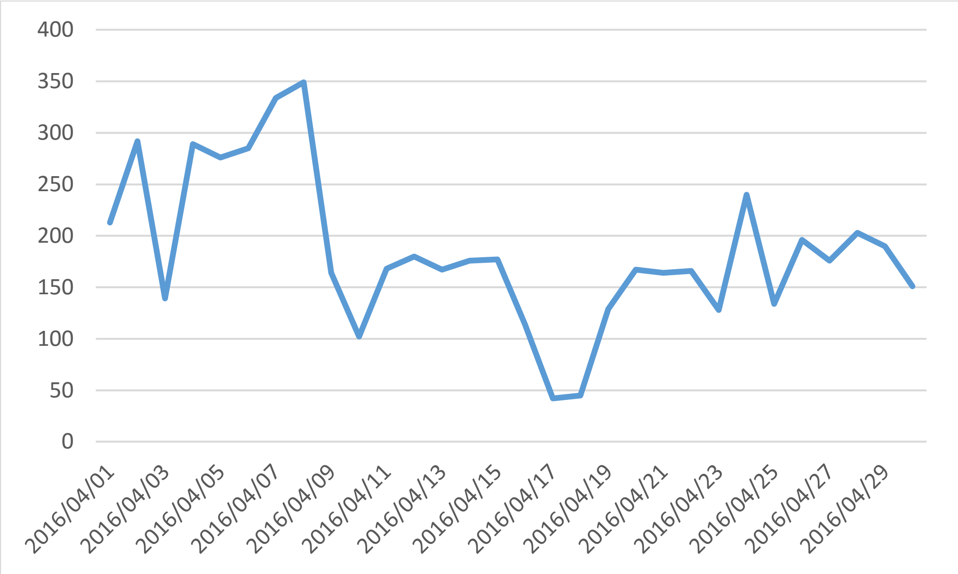

Figure 23: Redo log per day (GB) for T24LIVE1

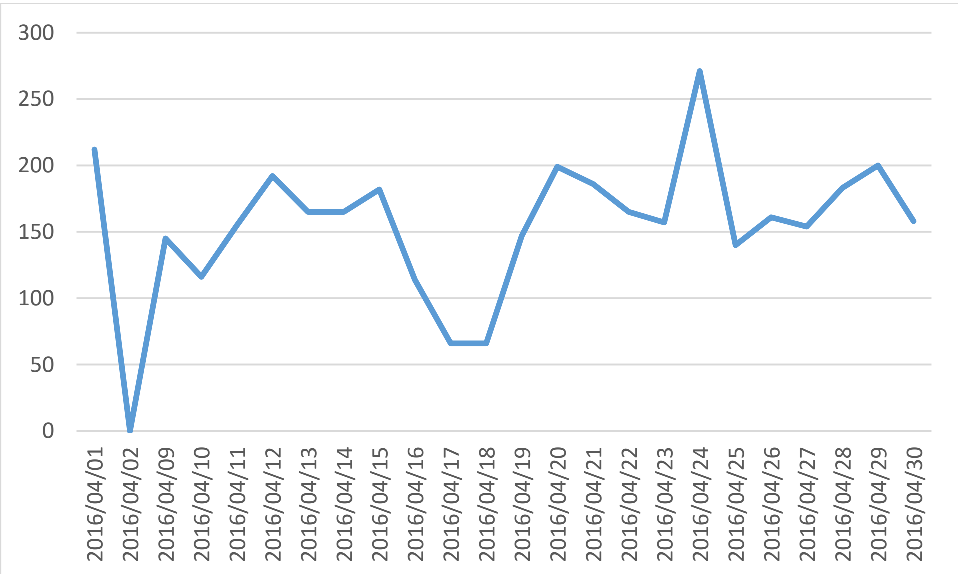

Figure 24: Redo log per day (GB) for T24LIVE2

5. Findings

5.1. Indexes with low distinct keys

These indexes have low distinct keys which makes queries that use them are not efficient:

OWNER	TABLE_NAME	INDEX_NAME	NUM ROWS	DISTINCT KEYS	BLEVEL
T24LIVE	FBNK_MM_M000	IX_FBNK_MM_M000_C7	26957219	8	2
T24LIVE	FBNK_INFO_CARD	IX_FBNK_INFO_CARD_CLOB_C76	4296744	3	2
T24LIVE	FBNK_INFO_CARD	NIX_FBNK_INFO_CARD_C8	4105302	6	2
T24LIVE	FBNK_ACCOUNT	IX_FBNK_ACCOUNT_C143	3580558	2	2
T24LIVE	FBNK_ACCOUNT	IX_FBNK_ACCOUNT_C163	3580558	2	2
T24LIVE	FBNK_ACCOUNT	NIX_FBNK_ACCOUNT_C163	3580558	1	2
T24LIVE	FBNK_QUANLYTHE	IX_FBNK_QUANLYTHE_CLOB_C23	2916500	8	2
T24LIVE	FBNK_QUANLYTHE	IX_FBNK_QUANLYTHE_CLOB_C34	2916500	9	2
T24LIVE	FBNK_VISA_OFFUS_TCB	IX_FBNK_VISA_OFFUS_TCB_C14	1611382	3	2
T24LIVE	FBNK_VISA_OFFUS_TCB	IX_FBNK_VISA_OFFUS_TCB_C25	1611382	2	2
T24LIVE	FBNK_FT_IN_PRINT_TCB	IX_FBNK_FT_IN_PRINT_TCB_C17	1444021	3	2
T24LIVE	FBNK_FT_IN_PRINT_TCB	IX_FBNK_FT_IN_PRINT_TCB_C18	1444021	3	2
T24LIVE	FBNK_LIMIT	NIX_FBNK_LIMIT_C84	1306044	1	2
T24LIVE	FBNK_LIMIT	IX_FBNK_LIMIT_C73	1209320	2	2
T24LIVE	FBNK_LIMIT	IX_FBNK_LIMIT_C84	1177906	4	2
T24LIVE	FBNK_DX_TRANSACTION	IX_FBNK_DX_TRANSACTION_C13	741963	6	2
T24LIVE	FBNK_DX_TRANSACTION	IX_FBNK_DX_TRANSACTION_C5	739796	3	2
T24LIVE	FBNK_TRICH_NO_TKTT_TCB	NIX_FBNK_TRICH_NO_TKTT_TCB_C17	523152	1	2
T24LIVE	FBNK_TRICH_NO_TKTT_TCB	NIX_FBNK_TRICH_NO_TKTT_TCB_C7	523152	1	2
T24LIVE	FBNK_MM_MONEY_MARKET	IX_FBNK_MM_MONEY_MARKET_C115	393746	2	2
T24LIVE	FBNK_MM_MONEY_MARKET	IX_FBNK_MM_MONEY_MARKET_C122	393746	3	2
T24LIVE	FBNK_MM_MONEY_MARKET	IX_FBNK_MM_MONEY_MARKET_C2	393746	8	2
T24LIVE	FBNK_MM_MONEY_MARKET	IX_FBNK_MM_MONEY_MARKET_C66	393746	3	2

T24LIVE	FBNK_MM_MONEY_MARKET	IX_FBNK_MM_MONEY_MARKET_C74	393746	2	2
T24LIVE	FBNK_MM_MONEY_MARKET	IX_FBNK_MM_MONEY_MARKET_C90	393746	2	2
T24LIVE	FBNK_MM_MONEY_MARKET	NIX_FBNK_MM_MONEY_MARKET_C129	393746	1	2
T24LIVE	FBNK_MM_MONEY_MARKET	NIX_FBNK_MM_MONEY_MARKET_C7	393746	7	2
T24LIVE	FBNK_MM_MONEY_MARKET	NIX_FBNK_MM_MONEY_MARKET_C8	393746	1	2
T24LIVE	FBNK_FUNDS_TRANSFER	NIX_FBNK_FUNDS_TRANSFER_C18	317544	4	2
T24LIVE	FBNK_COLLATERAL_RIGHT	IX_FBNK_COLLATERAL_RIGHT_C14	263644	3	2
T24LIVE	F_IB_USER	IX_F_IB_USER_C36	261082	1	2
T24LIVE	FBNK_PP_SERIES_TCB	IX_FBNK_PP_SERIES_TCB_C3	218279	7	2
T24LIVE	FBNK_QUAN003	NIX_FBNK_QUAN003_C10	211098	1	1
T24LIVE	FBNK_CHAR003	IX_FBNK_CHAR003_C15	205918	3	1
T24LIVE	FBNK_CHAR003	IX_FBNK_CHAR003_C17	205918	6	1
T24LIVE	FBNK_CHEQUE_ISSUE	NIX_FBNK_CHEQUE_ISSUE_C1	195047	1	1
T24LIVE	FBNK_PACIFIC_AIRLINES	NIX_FBNK_PACIFIC_AIRLINES_C18	184566	4	1
T24LIVE	FBNK_EVN_LOG_TCB	IX_FBNK_EVN_LOG_TCB_C11	179742	1	2
T24LIVE	FBNK_AUTO_TRANSFER_TCB	IX_FBNK_AUTO_TRANSFER_TCB_C5	174182	2	1
T24LIVE	FBNK_AUTO_TRANSFER_TCB	IX_FBNK_AUTO_TRANSFER_TCB_C19	170642	4	1
T24LIVE	F_FTBULK_CONTROL_TCB	IX_F_FTBULK_CONTROL_TCB_C28	147044	8	1
T24LIVE	FBNK_EVN_LOG_TCB	IX_FBNK_EVN_LOG_TCB_C7	126977	3	1
T24LIVE	FBNK_FTBULK_TCB	NIX_FBNK_FTBULK_TCB_C4	121228	5	2
T24LIVE	FBNK_FTBULK_TCB	NIX_FBNK_FTBULK_TCB_C5	121228	6	2
T24LIVE	FBNK_FTBULK_TCB	NIX_FBNK_FTBULK_TCB_C8	121228	7	2
T24LIVE	F_TSA_STATUS	IX_F_TSA_STATUS_C2	93895	4	2
T24LIVE	F_TSA_STATUS	IX_F_TSA_STATUS_C6	93895	8	1
T24LIVE	FBNK_GET_GIFT_TCB	IX_FBNK_GET_GIFT_TCB_C27	92168	3	1
T24LIVE	FBNK_LD_L000	IX_FBNK_LD_L000_C107	91577	1	1
T24LIVE	FBNK_LD_L000	IX_FBNK_LD_L000_C90	91360	3	1
T24LIVE	FBNK_LD_L000	IX_FBNK_LD_L000_C2	62580	7	1
T24LIVE	FBNK_LD_L000	NIX_FBNK_LD_L000_C13	62580	4	1
T24LIVE	F_EXCEPTION_LOG_HIST	NIX_F_EXCEPTION_LOG_HIST_C13	55145	3	2
T24LIVE	FBNK_EBAN000	IX_FBNK_EBAN000_C8	47079	4	1
T24LIVE	FBNK_AZ_ACCOUNT	NIX_FBNK_AZ_ACCOUNT_C107	33210	1	1
T24LIVE	FBNK_EB_COMPANY_CHANGE	IX_FBNK_EB_COMPANY_CHANGE_C1	24998	5	1
T24LIVE	FBNK_EB_COMPANY_CHANGE	NIX_FBNK_EB_COMPANY_CHANGE_C4	24998	1	1

T24LIVE	FBNK_DRAWINGS	NIX_FBNK_DRAWINGS_C14	20213	1	1
T24LIVE	FBNK_EB_ECM_XEPHANG	NIX_FBNK_EB_ECM_XEPHANG_C9	16912	2	1
T24LIVE	FBNK_FOREX	IX_FBNK_FOREX_C135	11203	1	1
T24LIVE	FBNK_QUANLYTHE#NAU	IX_FBNK_QUANLYTHE#NAU_C76	8492	2	1
T24LIVE	FBNK_POS_MVMT_TODAY	NIX_FBNK_POS_MVMT_TODAY_C25	8433	1	2
T24LIVE	FBNK_INFO_CARD#NAU	IX_FBNK_INFO_CARD#NAU_C137	7132	2	1

More index on table will make DML (INSERT, UPDATE, DELETE, MERGE ...) slower.

Recommendation: Drop or change these indexes to unusable state.

5.2. Optimize table FBNK_CATEG_ENT_ACTIVITY

Table FBNK_CATEG_ENT_ACTIVITY was created with script:

```
CREATE TABLE T24LIVE.FBNK_CATEG_ENT_ACTIVITY
(
  RECID      VARCHAR2(255 BYTE),
  XMLRECORD  BLOB
)
LOB (XMLRECORD) STORE AS SECUREFILE (
  TABLESPACE  DATAT24LIVE
  DISABLE     STORAGE IN ROW
  CHUNK       8192
  COMPRESS    HIGH
  CACHE
  NOLOGGING)
NOCOMPRESS
TABLESPACE DATAT24LIVE
```

Table size: 3.21 GB

Total size of LOBs: 299 GB

Number of rows: 36.342.930

Average length of XMLRECORD: 158

Because LOB was stored OUT of row, table size is larger than it should be.

After create with ENABLE STORAGE IN ROW, table size reduced to 8GB.

Recommendation: Recreate table with ENABLE STORAGE IN ROW option.

5.3. Avoid using literal value on conditions with high distinct values

Use literal values on column with high number of distinct values create a large number of entries in shared pool, takes longer time to process SQL.
These SQL should change to use bind variables:

DUPLICATE COUNT	EXECUTIONS	SQL TEXT
11048	42993	SELECT t.RECID FROM F_EXTERNA000 t WHERE RECID LIKE '20475873%' ESCAPE '\'
10282	30771	SELECT t.RECID,t.XMLRECORD.getClobVal() FROM F_IM_DOCUMENT_IMAGE t WHERE NVL(EXTRACTVALUE(t.XMLRECOR D,'/row/c4'),'^A')='22946348'
9582	24028	SELECT t.RECID FROM FBNK_AA_ARRANGEMENT t WHERE NVL(NUMCAST(EXTRACTVALUE(t.XMLRECORD,'/row/c1')),0)= 29816768
7009	13897	SELECT t.RECID,t.XMLRECORD.getClobVal() FROM FBNK_LD_L000 t WHERE NVL(NUMCAST(EXTRACTVALUE(t.XMLRECO RD,'/row/c11')),0)>=21050 and NVL(NUMCAST(EXTRACTVALUE(t.XMLRECORD,'/row/c11')),0)<=21074 and NVL(NU MCAST(EXTRACTVALUE(t.XMLRECORD,'/row/c1')),0)=20898889 and NVL(EXTRACTVALUE(t.XMLRECORD,'/row/c250') ,'^A')='VN0010319'
6733	74029	SELECT t.RECID FROM FBNK_PD_PAYMENT_DUE t WHERE RECID='PDLD1528643404'
6598	808555	SELECT t.RECID FROM F_OS_XML_CACHE t WHERE RECID LIKE 'ENQUIRY.RECORD.TOTAL_%_HOANKIEUTHI1009_%' ESCAPE '\'
4675	503653	SELECT t.RECID FROM FBNK_MERCHANT_TCB t WHERE NVL(EXTRACTVALUE(t.XMLRECORD,'/row/c9'),'^A')='1052000 3353011' or NVL(EXTRACTVALUE(t.XMLRECORD,'/row/c12'),'^A')='10520003353011'
4595	356085	SELECT t.RECID FROM F_RECORD_LOCK t WHERE NVL(EXTRACTVALUE(t.XMLRECORD,'/row/c2'),'^A')='DAHONG84'
3805	12811	SELECT t.RECID FROM F_AGENTCODE_TCB t WHERE RECID LIKE '%29052475%' ESCAPE '\'
3749	69152	SELECT t.RECID FROM FBNK_CUSTOMER_POSITION t WHERE RECID LIKE '%*VAN11.151*%' ESCAPE '\'
2382	4670	SELECT t.RECID FROM FBNK_DEPO_WITHDRA t WHERE NVL(EXTRACTVALUE(t.XMLRECORD,'/row/c29'),'^A')='201605 04' and NVL(EXTRACTVALUE(t.XMLRECORD,'/row/c34'),'^A')='VN0010326'
1151	4228503	SELECT ex_or_include, parameters FROM USER_XML_INDEXES WHERE TABLE_NAME = 'F_VERSION'
1110	3608	SELECT t.RECID FROM FBNK_PD_BALANCES t WHERE RECID LIKE 'PDLD1608661990%' ESCAPE '\'
1022	1902	SELECT t.RECID FROM FBNK_LMM_SCHEDULES_PAST t WHERE RECID LIKE 'MM1609502486%' ESCAPE '\' ORDER BY N UMSORT(RECID)
1000	1400	SELECT t.RECID FROM FBNK_INFO_CARD#NAU t WHERE NVL(EXTRACTVALUE(t.XMLRECORD,'/row/c2'),'^A')='197211 74268010'

Recommendation: Use bind variables on condition with high number of distinct values.

5.4. Partition table with high concurrency access

Table F_JOB_LIST_78 is taking a large number of "buffer busy wait" event. This also contributes to transfer through interconnect network.

Top Buffer Busy Waits segments

Owner	Object Name	Obj. Type	Buffer Busy Waits	% of Capture
T24LIVE	F_JOB_LIST_78	TABLE	50,895	43.99

T24LIVE	LOB_F_JOB_LIST_78	LOB	10,701	9.25
T24LIVE	SYS_IL0000298291C00002$$	INDEX	7,381	6.38
T24LIVE	FBNK_FUNDS_TRANSFER#HIS	TABLE PARTITION	2,592	2.24
T24LIVE	FBNK_IC_C015	TABLE	2,218	1.92

Top Global Cache Buffer Busy

Owner	Object Name	Obj. Type	GC Buffer Busy	% of Capture
T24LIVE	F_JOB_LIST_78	TABLE	216,535	53.92
T24LIVE	SYS_IL0000298291C00002$$	INDEX	25,754	6.41
T24LIVE	LOB_F_JOB_LIST_78	LOB	24,698	6.15
T24LIVE	FBNK_FUNDS_TRANSFER#HIS_PAR_PK	INDEX	13,490	3.36
T24LIVE	F_BATCH_STATUS	TABLE PARTITION	12,969	3.23

Recommendation: Partition table F_JOB_LIST_78 by HASH to reduce "buffer busy wait" event. Also partition all F_JOB_LIST_xx tables to prevent from further issues.

5.5. Optimize enq: HW - contention wait event

During COB time, wait event "*enq: HW - contention*" takes a large number of DB time:

Event	Waits	Total Wait Time (sec)	Wait Avg(ms)	% DB time	Wait Class
db file sequential read	1,376,377	7418.2	5	22.8	User I/O
DB CPU		6808.5		21.0	
enq: HW - contention	18,778	6549.6	349	20.2	Configuration
log file sync	903,900	3664	4	11.3	Commit

This event related to LOB segments extent.
SQL affect mostly with this event:

```
MERGE INTO FBNK_LD_L000 USING DUAL ON (RECID = :RECID)
WHEN MATCHED
THEN UPDATE SET XMLRECORD = XMLTYPE ( :XMLRECORD, NULL, 1, 1)
WHEN NOT MATCHED
THEN INSERT (XMLRECORD, RECID) VALUES (XMLTYPE ( :XMLRECORD, NULL, 1, 1),
:RECID)
```

Set this event will make some improvement:
ALTER SYSTEM SET EVENT = '44951 TRACE NAME CONTEXT FOREVER, LEVEL 1024' SCOPE=SPFILE;

Refer to MOS note Analyzing 'enq: HW - contention' Wait Event (Doc ID 740075.1) for more infomation.

The parameter is also recommended by Temenos.

Recommendation: Set the EVENT parameter & restart DB.

5.6. High SQ Enqueue Contention with LOBs

We observe there is high SQ enqueue contention on the system.

Event	Waits	Total Wait Time (sec)	Wait Avg(ms)	% DB time	Wait Class
db file sequential read	1,828,243	10K	5	28.7	User I/O
DB CPU		8362.8		23.9	
log file sync	998,226	4802	5	13.7	Commit
enq: HW - contention	30,349	2254.3	74	6.4	Configuration
enq: SQ - contention	1,326	1476.5	1114	4.2	Configuration

With system has high volume of transactions on lob columns, the cache size for IDGEN1$ has a default of 1000, which is probably too small.

We should increase to 10000 to remedy this issue:

SQL> ALTER SEQUENCE SYS.IDGEN1$ CACHE 10000;

Recommendation: Set IDGEN1$ sequence cache to 10000. See Doc ID 432508.1 for more details.

Appendix 1 – Other Documentation

MTS and Large Pool:

Note:62140.1	Fundamentals of the Large Pool
Note:268581.1	Obsolete / Deprecated Initialization Parameters in 10G

Checkpoints:

Note:265831.1	Automatic Checkpoint Tuning in 10g
Note:274264.1	REDO LOGS SIZING ADVISORY in 10g

Statistics:

Note: 266040.1	Automatic statistics Gathering in oracle 10G
Note: 252597.1	Relation between Table Monitoring and STATISTICS_LEVEL parameter in 10g
Note:281790.1	Oracle Database 10g DBMS_STATS Package FORCE argument
Note: 283890.1	Oracle Database 10g Locking Statistics

Locally Managed Tablespaces:

Note:93771.1	Introduction to Locally-Managed Tablespaces
Note:262472.1	10g: BIGFILE Type Tablespaces Versus SMALLFILE Type

Tuning CPU Resources

Note: 33824.1	Statistic - recursive cpu usage
Note: 164768.1	Diagnosing High CPU Utilization
Note: 33828.1	Statistic - cpu used by this session (Reference Note)
Note: 33854.1	Statistic - parse time elapsed (Reference Note)
Note: 33853.1	Statistic - parse time cpu (Reference Note)
Note: 33828.1	Statistic - cpu used by this session (Reference Note)
Note: 276103.1	PERFORMANCE TUNING USING 10g ADVISORS AND MANAGEABILITY FEATURES

MTS and Large Pool:

Tuning I/O

Note: 30286.1	I/O Tuning with Different RAID Configurations
Note: 1037322.6	WHAT IS THE DB_FILE_MULTIBLOCK_READ_COUNT PARAMETER
Note: 148342.1	Avoiding I/O Disk Contention
Note: 245055.1	Oracle Database 10g Enhanced wait model
Note: 272360.1	Tablespace Groups for SQL Operations in 10g
Note: 242090.1	10g NEW FEATURE on SEGMENT SHRINK

Optimizing SQL Statements

Note: 10585.1	Query and Application Tuning using Explain and TKProf
Note: 163563.1	START POINT: My Query runs slowly
Note: 33089.1	TROUBLESHOOTING GUIDE: SQL Tuning
Note: 67522.1	Why is my index not used?
Note: 69992.1	Why is my hint ignored?
Note: 34558.1	Waitevent - db file scattered read (Reference Note)
Note: 34396.1	Waitevent - SQL*Net message from dblink (Reference Note)
Note: 34559.1	Waitevent - db file sequential read (Reference Note)
Note 259188.1	Oracle10g: Using SQLAccess Advisor (DBMS_ADVISOR) with the Automatic Workload Repository
Note: 262687.1	How to use the Sql Tuning Advisor
Note:244192.1	10g NEW FEATURE Automatic Database Diagnostic Monitor (ADDM)
Note:250655.1	How to use the Automatic Database Diagnostic Monitor
Note: 290027.1	Computationally intensive PL/SQL programs run fast on 10G as compared to 9i

Tuning Network Resources

Note: 44694.1	SQL*Net Packet Sizes (SDU and TDU Parameters)
Note: 1005123.6	Tuning SQL*Net for better performance

MTS and Large Pool:

Tuning Memory

Note: 257643.1	Oracle Database 10g Automated SGA Memory Tuning
Note: 295626.1	How To Use Automatic Shared Memory Management (ASMM) In Oracle10g

Appendix 2 - Methodology

A number of tools and methods were involved in examining and reviewing the systems, and providing the recommendations:

Statspack

Statspack is an Oracle provided utility that collects information and stores the performance statistics data permanently in Oracle tables, which can later be used for reporting and analysis. The data collected can be analyzed using the report provided, which includes an "instance health and load" summary page, high resource SQL statements, as well as the traditional wait events and initialization parameters.

High-Water Mark Viewer

This Excel spreadsheet connects to a local Oracle database via OO4O and extracts high-water mark information on all tables. It then generates an Excel chart that depicts the impact on full table scans.

Oracle Trender

This Excel spreadsheet connects to a local Oracle database via OO4O and extracts information about wait events, statistics etc. Several Excel charts are automatically generated that depict the impact of these events on the overall system.

KM Repository on Oracle Support

KM Repository on Oracle Support is Oracle's electronic delivery of support information. A premier service for all Oracle-supported customers, it has a wealth of white papers, technical bulletins, user forums, Oracle documentation, and is used to open and work Technical Assistance Requests, verify supported configurations, etc. http://support.oracle.com

SAR

SAR is a performance data collection program found on most Unix platforms. It is configurable for interval and duration, and can capture critical CPU and disk performance data.

Remote Diagnostic Agent

The Remote Diagnostic Agent is an Oracle-developed tool that is designed to collect significant amounts of configuration information from both hosts and databases. Primarily used to diagnose problem issues, the information can also be used proactively.

Appendix 3 - Caveats

There are several potential issues with a health check of this nature.

- The data held internally in SYS or SYSTEM tables and views can be tainted by issues that are now rectified. For example, a database has been running for 4 weeks with a frequently executed query resulting in an expensive full table scan against a 1 million block table. Just prior to the health check a unique index was added to this table resulting in all queries using a unique index lookup. V$SYSTEM_EVENT would probably still show that too many db file scattered read' waits had occurred, and the original queries against that table are likely in the top x queries ordered by buffer gets/physical reads, yet the issue has now been rectified.

- Several of the findings in this document are based on ratios. These are not always a solid basis for analyzing a particular component's performance. Using the frequently quoted buffer cache example, a database can have 5 sql statements that constitute 90% of the load. These queries result in full table scans of a relatively large table and are executed frequently so that each full table scan results in mostly logical rather than physical I/Os. The outcome of this scenario might be that the Buffer Cache Hit Ratio is a very nice 99.99%. Say changing the statements to use indexes and altering the application to execute the sql less often resulted in an 80% drop in overall I/Os. Likely the Buffer Cache Hit Ratio has now dropped but the database is much healthier as a result!

- Most importantly, the Performance Review is based on the difference between two snapshots of how the database has been performing at particular points in time, based on its load and sql at those times. Reducing the load on the database, for example by reducing logical and physical I/O, or reducing parsing, may negate the need to implement some of the recommendations outlined below. To give an example, the Shared Pool Advisor may be indicating that the shared pool size needs to be increased. However, changing the code to use bind variables instead of literals may increase cursor shareability and therefore free space in the shared pool. Result, shared pool no longer needs resizing so the recommendation now becomes redundant.

- In some cases the 'av rd (ms)' columns in a statspack report can show unusually high numbers - in some cases this can be converted to much more time than there is between snapshots. If this is the case in your existing statspack snapshots, you may need to apply patch 4942939.