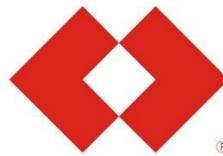




## *Advanced Customer Services*

# Report of Findings For TWO

Prepared For



**TECHCOMBANK**

### **Prepared by Advanced Customer Services**

Author: Do Duc Thinh

Creation Date: 24 May 2016

Version: 1.0

Data Collection: From 20-APR-16 02:00 to 22-APR-16 23:00

**ORACLE**

---

# 1. Contents

1. Contents .....	2
2. Executive Summary .....	3
3. System Reviewed .....	4
3.1. Host .....	4
3.2. Database .....	4
4. Overview .....	5
4.1. Database Response Time .....	5
4.2. Connection Balance .....	6
4.3. Average Active Session .....	7
4.4. CPU Utilization .....	8
4.5. Disk IO Activity .....	10
4.6. Logical IO .....	11
4.7. Transactions Rate and User Calls .....	12
4.8. PGA Memory Statistics .....	14
4.9. Redo Transaction Activity .....	17
5. Findings .....	19
5.1. Disable constraint .....	19
5.2. Resource Intensive SQL .....	19
5.2.1. SQL ID 335w4t4un0gzq - Module flora4.exe .....	19
5.2.2. SQL ID 17nkd3cxvmr7u - Module JDBC Thin Client .....	20
Appendix 1 - Other Documentation .....	22
Appendix 2 - Methodology .....	25
Appendix 3 - Caveats .....	26

---

## 2. Executive Summary

Oracle Advanced Customer Services (ACS) conducted a technical assessment on the Techcombank TWO database hosted on Windows hosts named dc-card-db01, dc-card-db02. 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 TWO 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 20-APR-16 02:00 to 22-APR-16 23:00.

Category	Information	Additional Comments
Database	Node 1 has double number of session compare to node 2.	Verify connection string from application & stop/start application to balance workload if possible
	Enable validate or drop DISABLED constraints	See <a href="#">Disable constraint</a>
	Review SQL with high resource usage	See <a href="#">Resource Intensive SQL</a>

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	dc-card-db01, dc-card-db02
Running DB Instances	1
Purpose (Production, development, Q&A)	Production
Platform	Microsoft Windows x86 64-bit
Model	
Operating System	Windows
O/S Version and Release	Windows Server 2008 R2
Cluster Software / Version	Grid Infrastructure 11.2.0.4
# CPU	12
Processor / CPU Speed	
Memory	95 GB
Volume Manager / Version	ASM

### 3.2. Database

Database Name	TWO
Instance Name	two1, two2
RAC-Configuration	Yes
Machine Name	dc-card-db01, dc-card-db02
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: 162 GB Tempfiles: 20 GB
Archiving Enabled?	Yes

---

## 4. Overview

Unless otherwise noted, all findings are based on data collected from 20-APR-16 02:00 to 22-APR-16 23:00.

### 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.

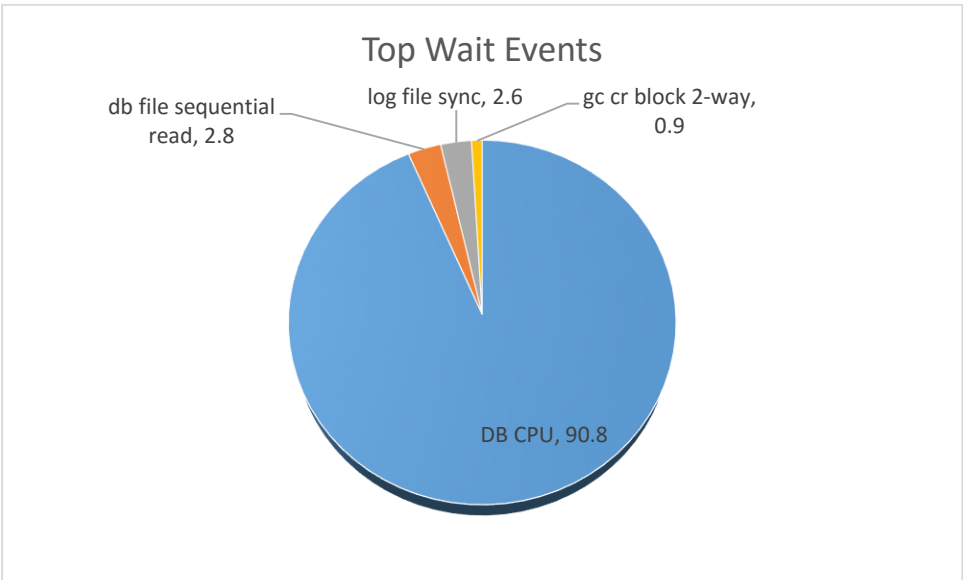
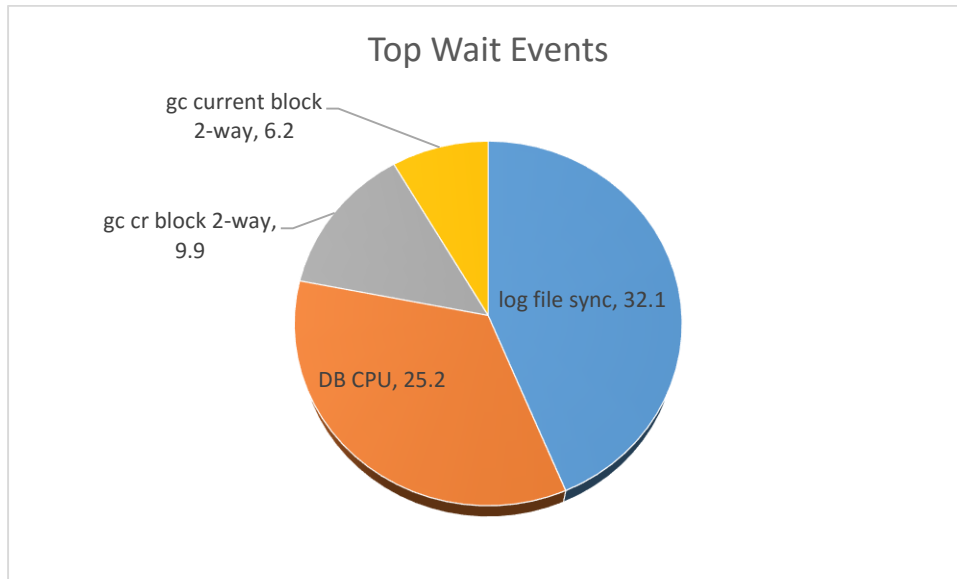


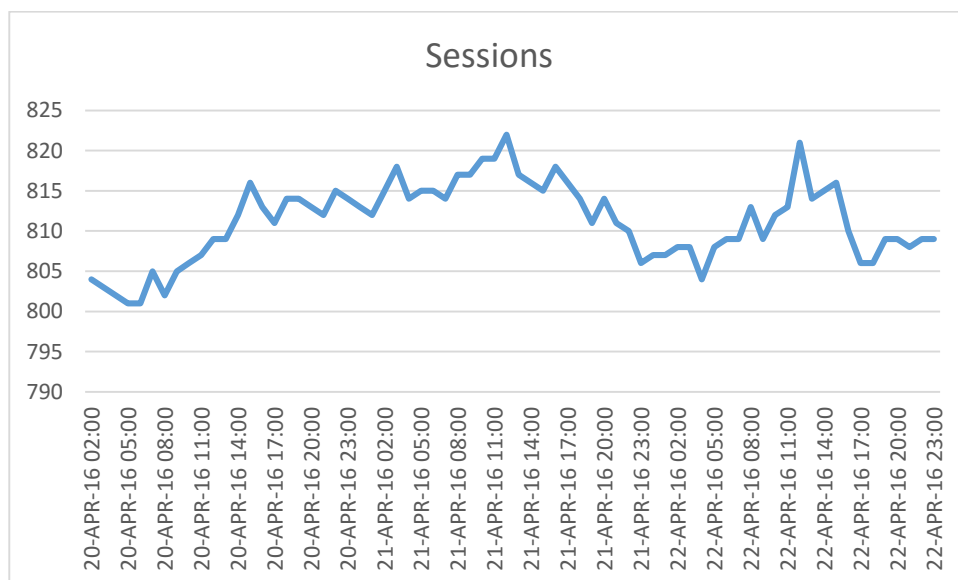
Figure 1: Top Foreground Waits for TWO1



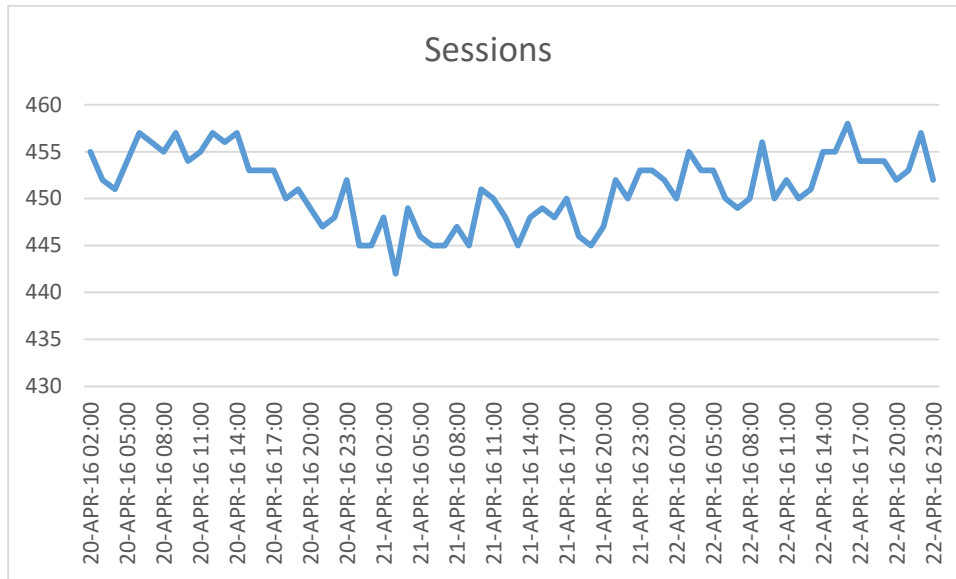
**Figure 2: Top Foreground Waits for TWO2**

## 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 TWO1**

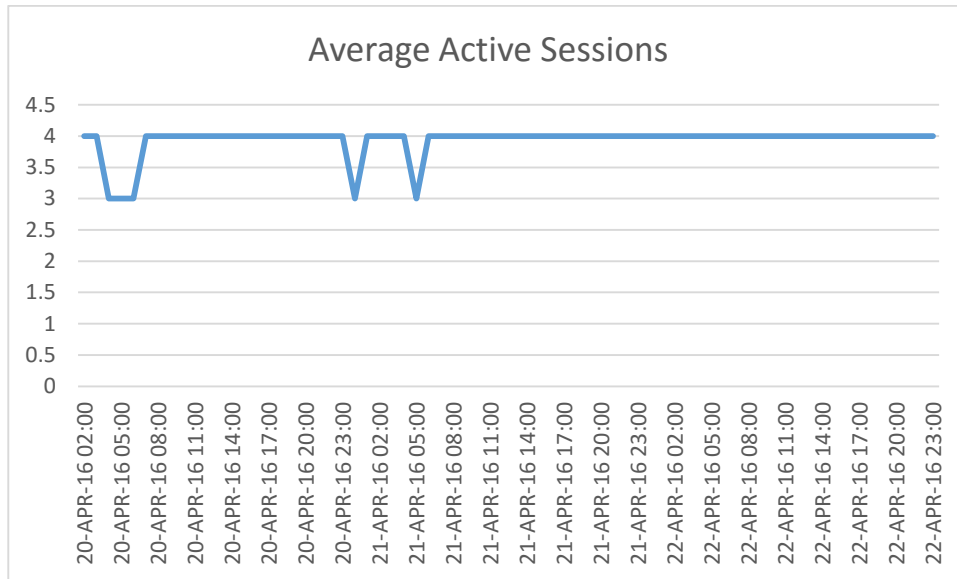


**Figure 4: Average Database Sessions for TWO2**

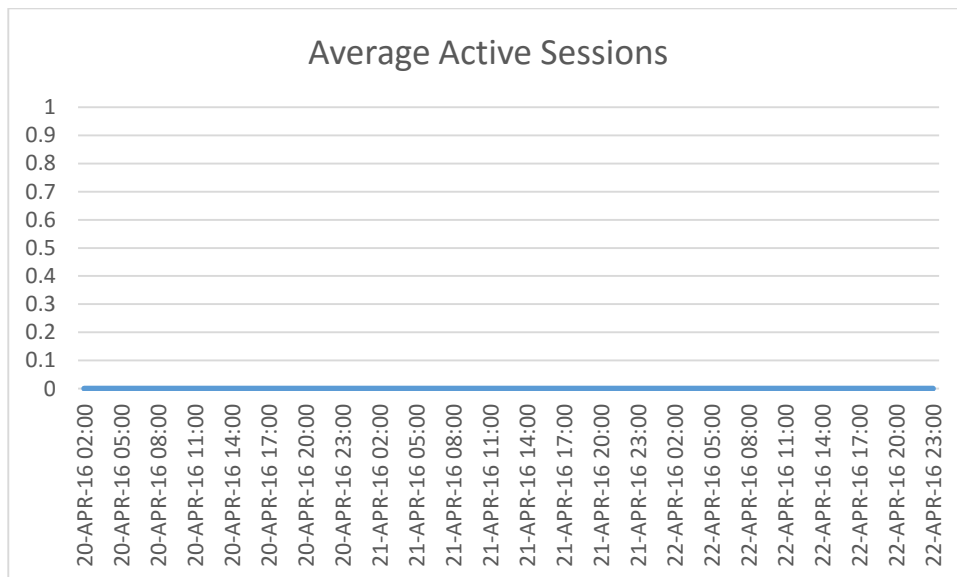
**Observation:** As seen from the data, average number of session is evenly distributed on both the instances. No connection spikes or logon/logoff storms are observed. This indicates that connections are successfully load balanced across all nodes in the cluster.

### 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 TWO1**



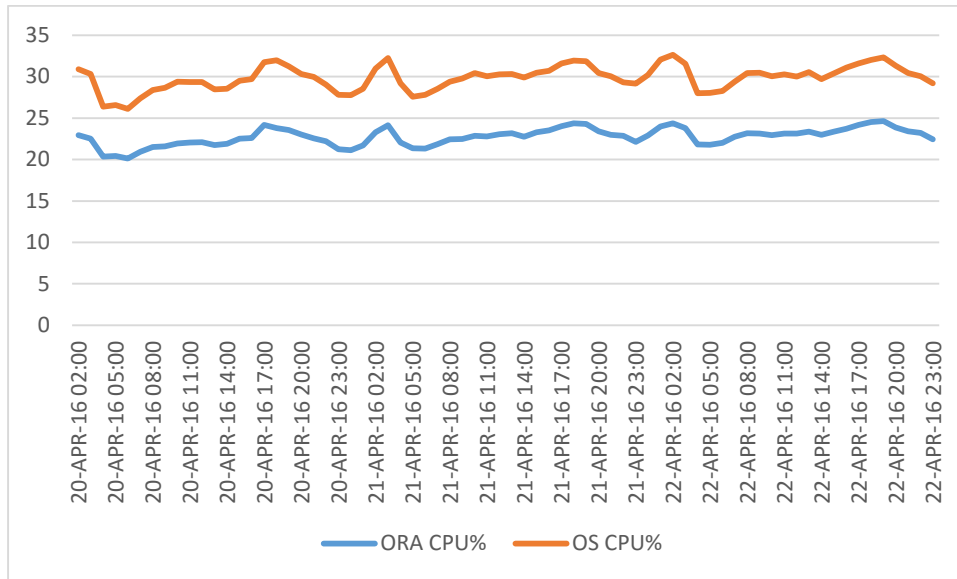
**Figure 6: Average Active Sessions for TWO2**

**Observation:** Average active session is within system capacity

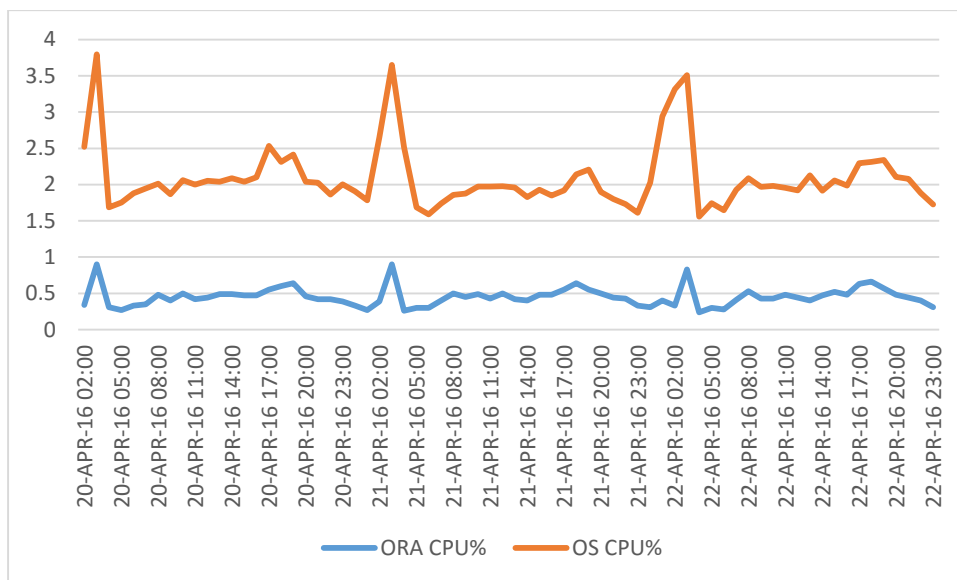
#### 4.4. CPU Utilization

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 TWO1**



**Figure 8: CPU Break Down for TWO2**

**Observation:**

The CPU utilization is well under server capacity. Oracle CPU is low compare to OS CPU. Verify which non-Oracle processes consume this CPU & reduce if possible.

## 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.

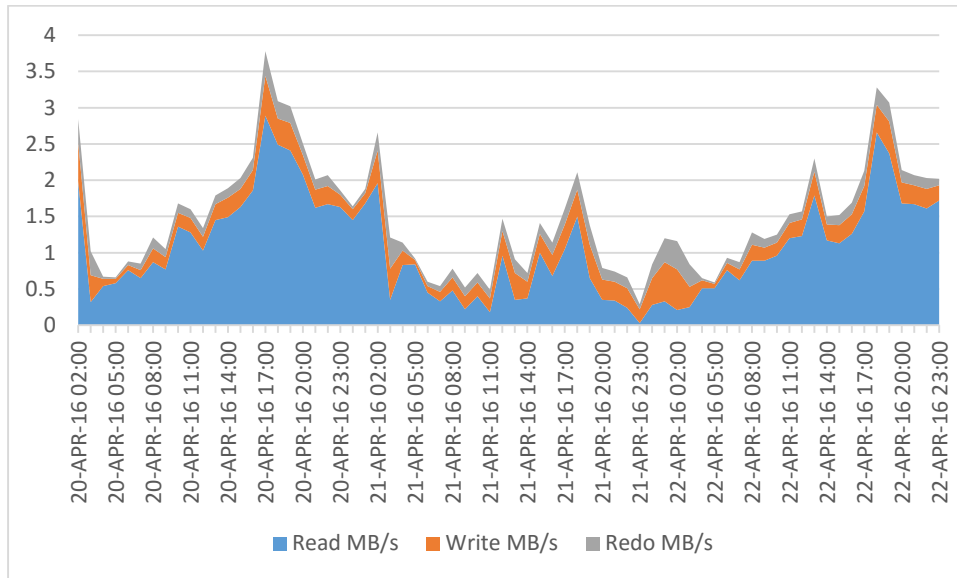


Figure 9: Disk IO for TWO1

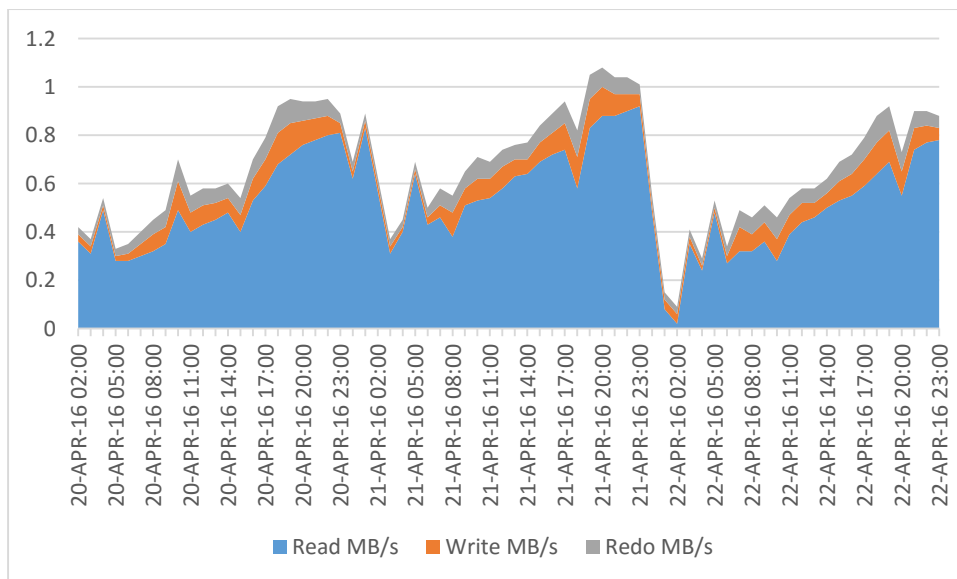
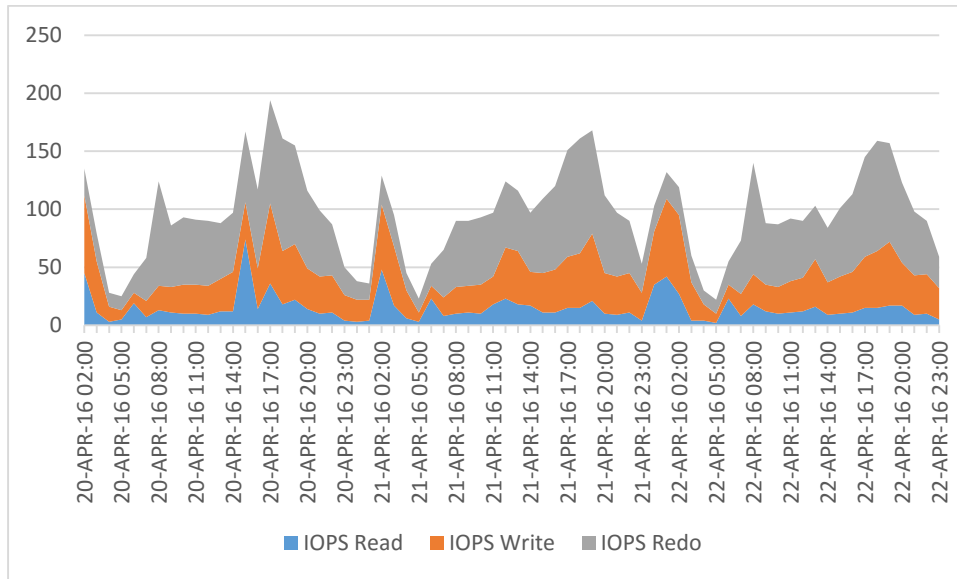
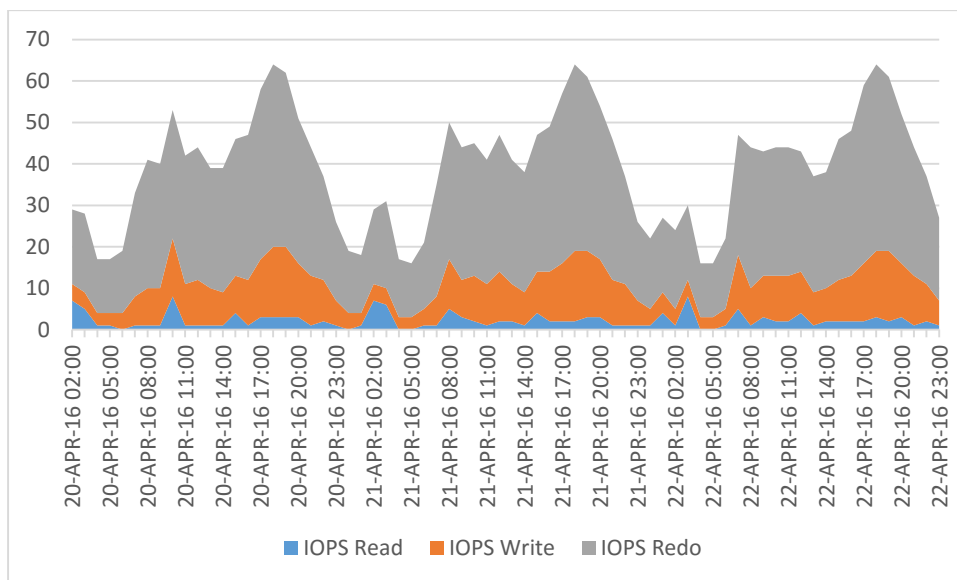


Figure 10: Disk IO for TWO2



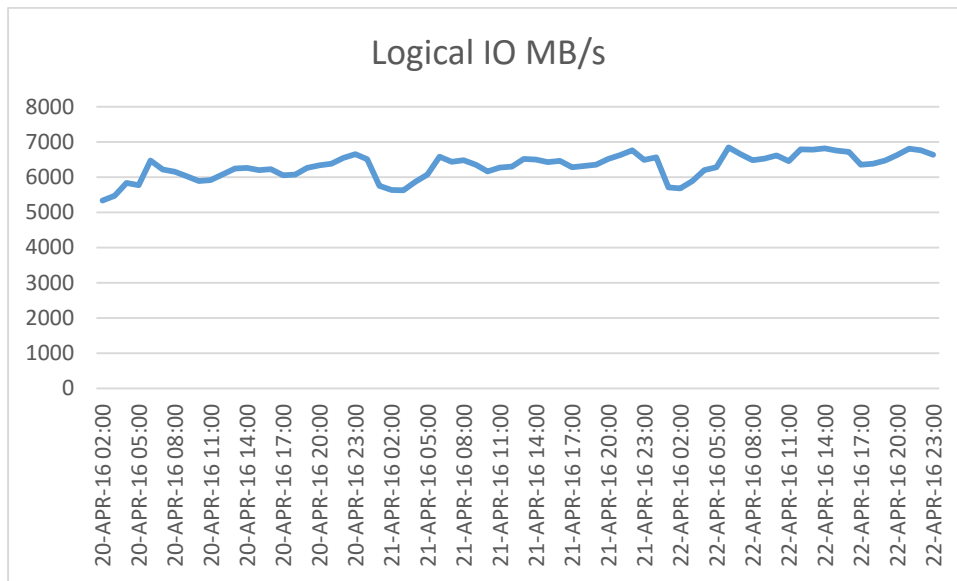
**Figure 11: Read , Write Operations per Second (IOPS) for TWO1**



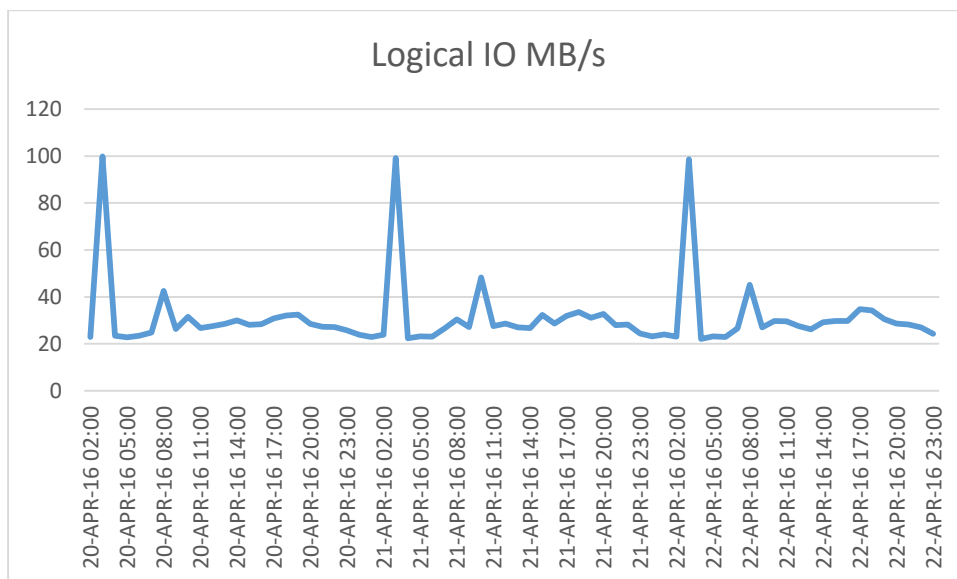
**Figure 12: Read , Write Operations per Second (IOPS) for TWO2**

#### 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.



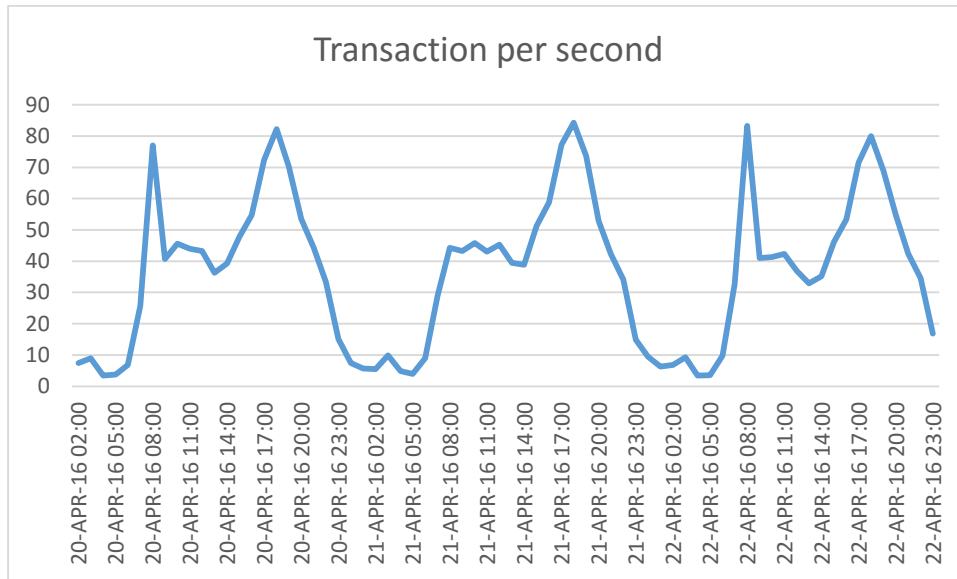
**Figure 13: Logical IO for TWO1**



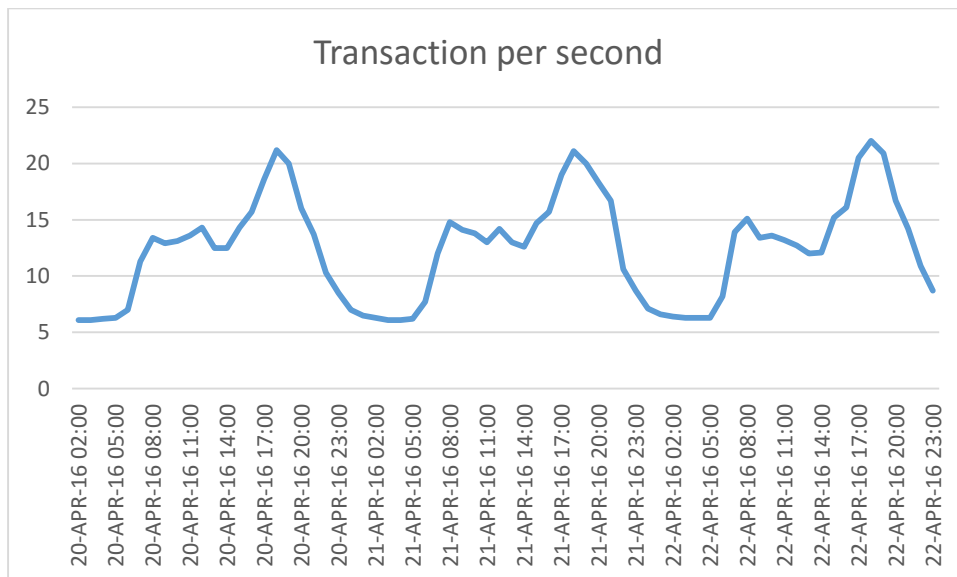
**Figure 14: Logical IO for TWO2**

#### 4.7. Transactions Rate and User Calls

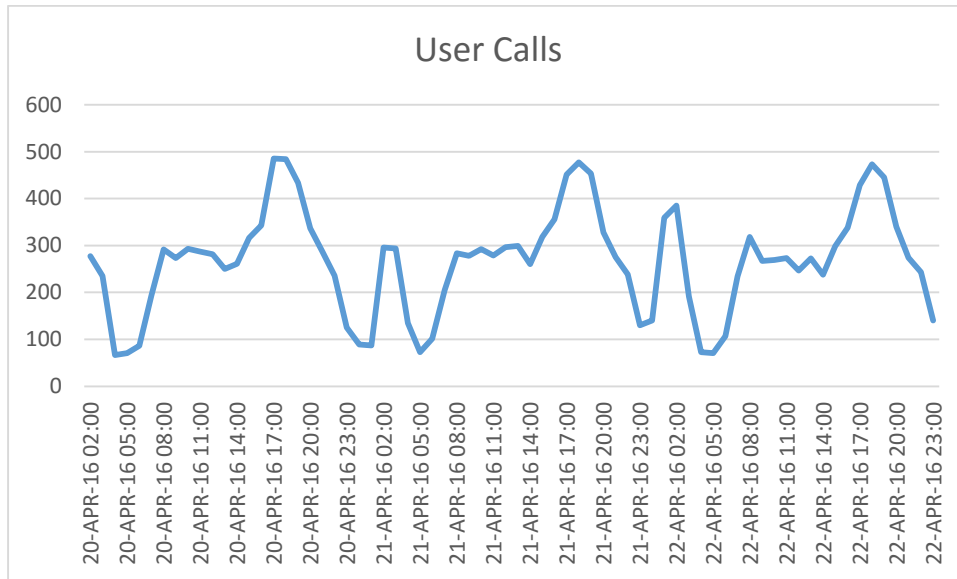
Peak periods are often defined by the transaction arrival rate. In Oracle, a transaction is defined by a series of operations that result in either a COMMIT operation, or a ROLLBACK operation.



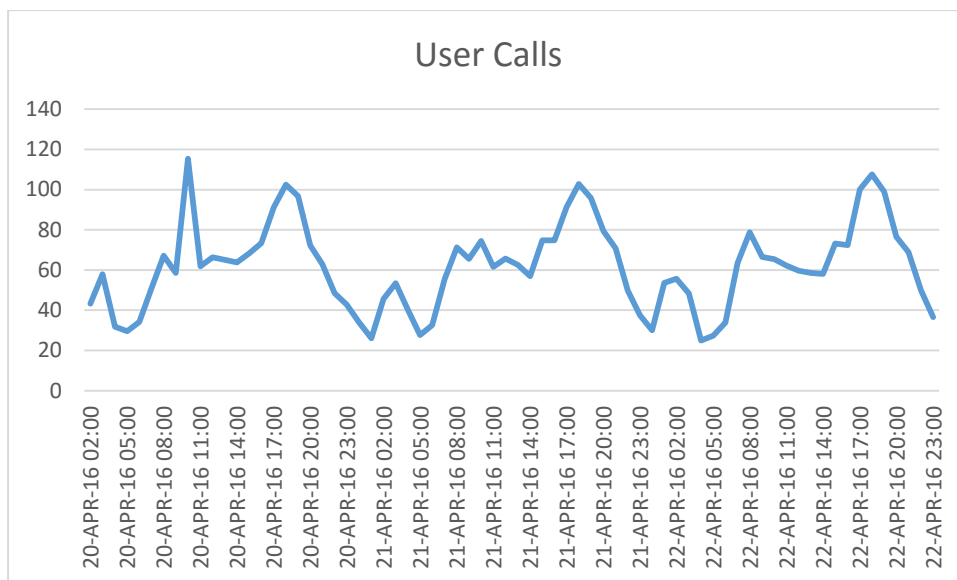
**Figure 15: Transactions per Second for TWO1**



**Figure 16: Transactions per Second for TWO2**



**Figure 17: User call per seconds for TWO1**



**Figure 18: User call per seconds for TWO2**

#### 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

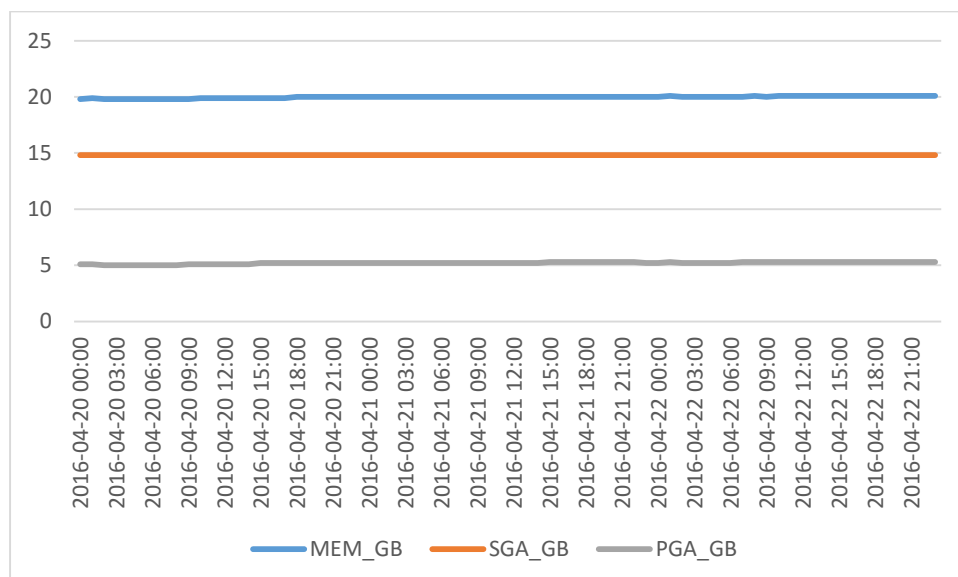
code acting on behalf of it. An example of such information is the runtime area of a cursor. Each time a cursor is executed, a new runtime area is created for that cursor in the PGA memory region of the server process executing that cursor. Analyze this section helps when using the new model to allocate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

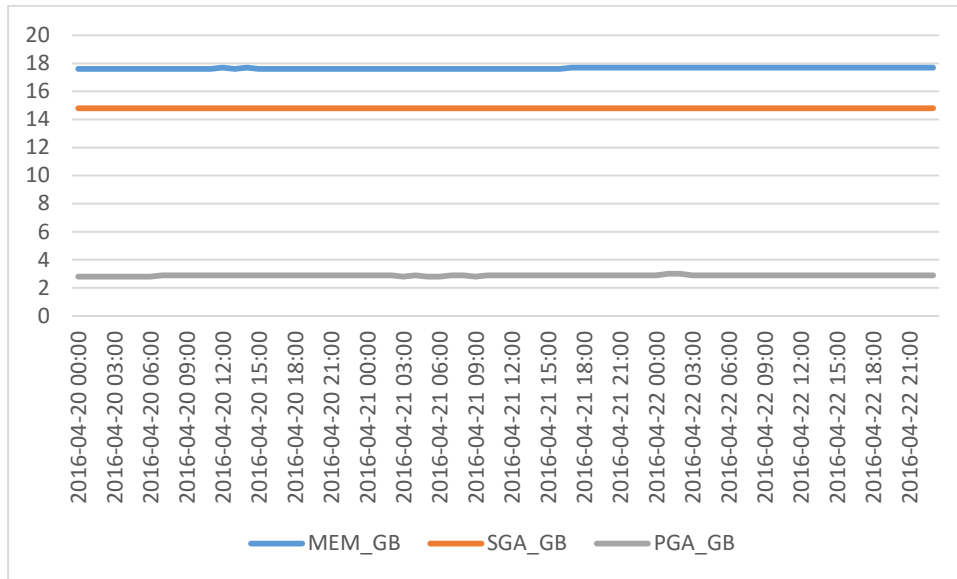
Under automatic PGA memory management mode, Oracle honors the PGA\_AGGREGATE\_TARGET limit by controlling dynamically the amount of PGA memory allotted to SQL database areas. At the same time, Oracle maximizes the performance of all the memory-intensive SQL operators by maximizing the number of database 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.

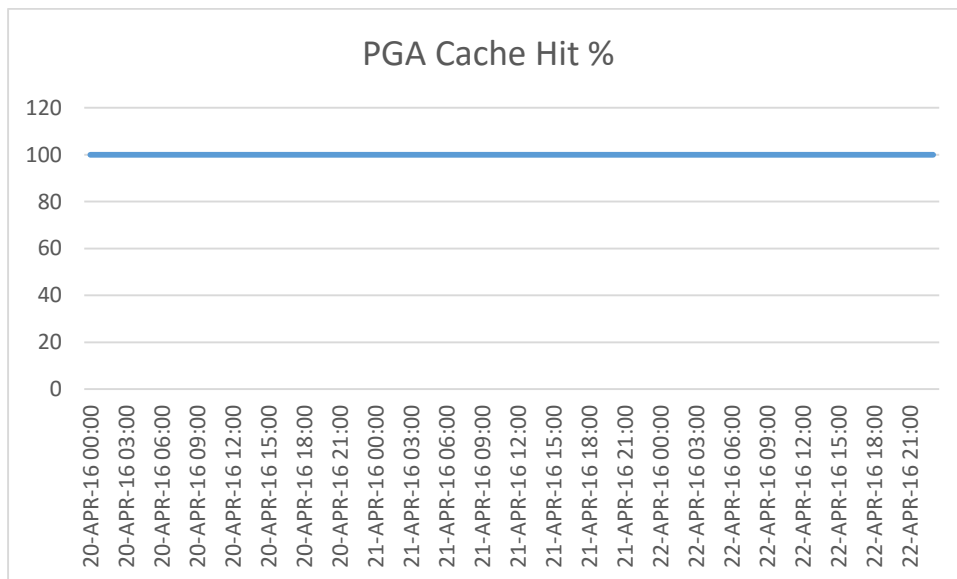
This metric is computed by Oracle to reflect the performance of the PGA memory component. It is cumulative from instance start-up. A value of 100% means that all work areas executed by the system since instance start-up have used an optimal amount of PGA memory. This is, of course, ideal but rarely happens 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 TWO1**

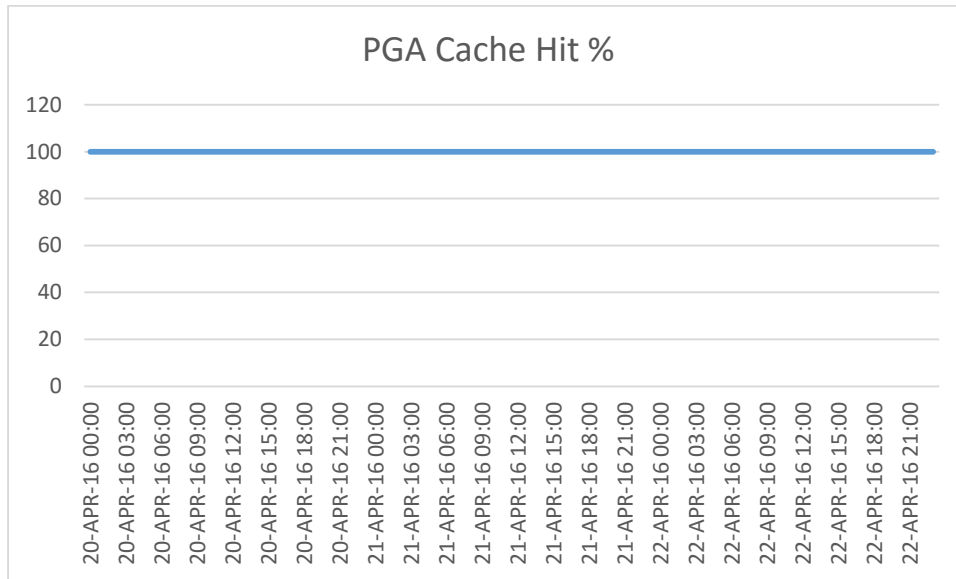


**Figure 20: Memory allocated for TWO2**



**Figure 21: PGA cache hit % for TWO1**





**Figure 22: PGA cache hit % for TWO2**

**Observation:**

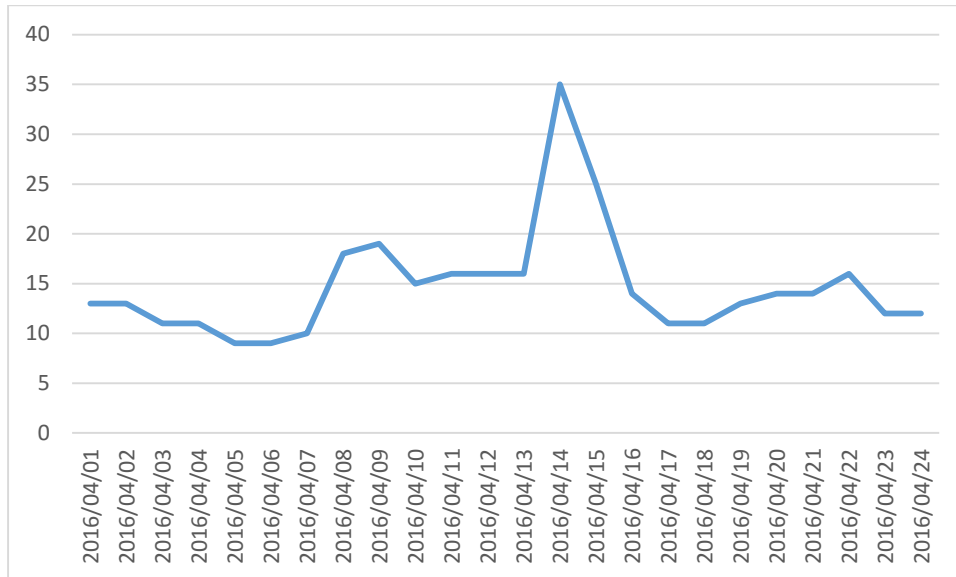
The current size of PGA is already optimal.

#### 4.9. Redo Transaction Activity

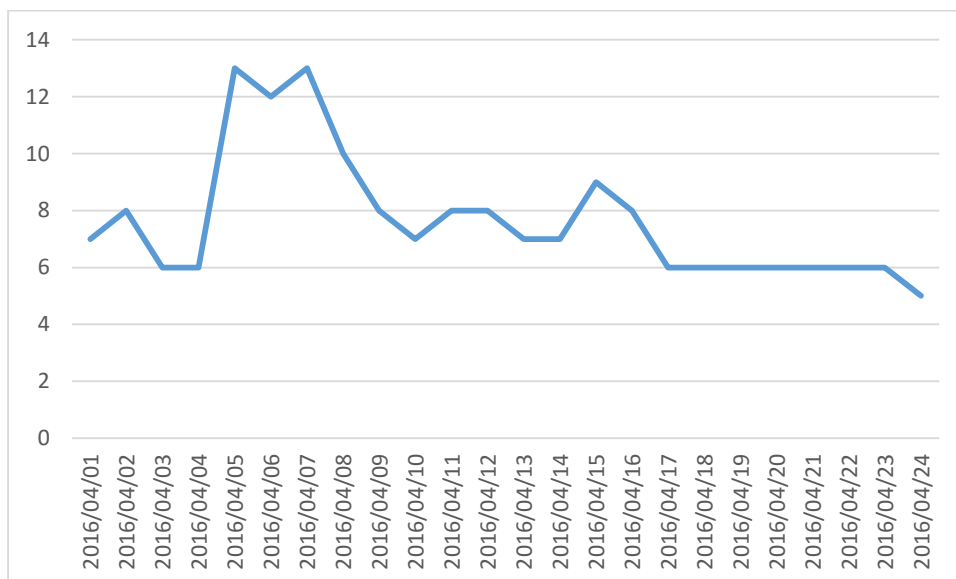
Redo logs contain the transaction data that is created when users submit changes to the database. As each redo log fills to capacity with these changes, it “switches” to the next redo log to continue, while the just filled redo log is then copied by the archiving process. If redo log switching is occurring 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	1	+DATA01/two/onlineolog/group_1.257.892564839	1
2	1	+FRA01/two/onlineolog/group_1.258.892564841	1
3	2	+DATA01/two/onlineolog/group_2.256.892564843	1
4	2	+FRA01/two/onlineolog/group_2.257.892564845	1
5	3	+DATA01/two/onlineolog/group_3.280.894844271	1
6	3	+FRA01/two/onlineolog/group_3.267.894844273	1
7	11	+DATA01/two/onlineolog/group_11.259.894843693	1
8	11	+FRA01/two/onlineolog/group_11.366.894843695	1
9	12	+DATA01/two/onlineolog/group_12.283.894843765	1
10	12	+FRA01/two/onlineolog/group_12.431.894843765	1

11	13	+DATA01/two/online/onlinegroup_13.285.894843771	1
12	13	+FRA01/two/online/onlinegroup_13.451.894843773	1



**Figure 23: Redo log per day (GB) for TWO1**



**Figure 24: Redo log per day (GB) for TWO2**

---

## 5. Findings

### 5.1. Disable constraint

Constraints are used for data integrity and consistency. Disabled constraints can lead to inconsistent data.

#	OWNER	CONSTRAINT_NAME	TABLE_NAME	STATUS	LAST_CHANGE
1	LOYALTWO	SYS_C0020437	SMS_CREDATA	DISABLED	2013-07-28/02:54:26
2	LOYALTWO	SYS_C0020438	SMS_CREDATA	DISABLED	2013-07-28/02:54:26
3	LOYALTWO	SYS_C0020439	SMS_CREDATA	DISABLED	2013-07-28/02:54:26

**Recommendation:** Enable validate or drop these constraints.

### 5.2. Resource Intensive SQL

Any effort of improving performance must involve reviewing the actual SQL code that is submitted in transactions to determine if it can be optimized. The largest performance improvements usually come from tuning the actual code, not from adjusting database parameters. It is important to periodically extract the most resource intensive SQL and examine it for improvement opportunities.

Following SQL statements should be tuned to gain maximum benefits.

#### 5.2.1. SQL ID 335w4t4un0gzq - Module flora4.exe

```
SELECT COUNT (*)
FROM TLG T
WHERE      T.PAN = :B1
          AND T.TRANCODE = 10
          AND T.RESPCODE = 1
          AND TYPE <> 420
          AND REVREQUESTID IS NULL
          AND T.TIME >= SYSDATE - 1
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT				5 (100)
1	SORT AGGREGATE		1	33	
2	TABLE ACCESS BY INDEX ROWID	TLG	1	33	5 (0)
3	INDEX RANGE SCAN	IDX_TLG_TIME	1		3 (0)

Stat Name	Statement Total	Per Execution	% Snap Total
Elapsed Time (ms)	6,448,244	2,518.85	0.65
CPU Time (ms)	4,887,168	1,909.05	0.54
Executions	2,560		
Buffer Gets	2,540,345,751	992,322.56	1.23
Disk Reads	26,296	10.27	0.07
Parse Calls	8	0.00	0.00
Rows	2,560	1.00	

Access by index IDX\_TLG\_PAN is more effective as it can filter more rows:

```
SELECT /*+ index (T, IDX_TLG_PAN */ COUNT (*)
FROM TLG T
WHERE T.PAN = :B1
      AND T.TRANCODE = 10
      AND T.RESPCODE = 1
      AND TYPE <> 420
      AND REVREQUESTID IS NULL
      AND T.TIME >= SYSDATE - 1
```

Plan hash value: 1822208071

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	33	7 (0)	00:00:01
1	SORT AGGREGATE		1	33		
* 2	TABLE ACCESS BY INDEX ROWID	TLG	1	33	7 (0)	00:00:01
* 3	INDEX RANGE SCAN	IDX_TLG_PAN	6		3 (0)	00:00:01

Predicate Information (identified by operation id):

- ```

2 - filter("T"."TRANCODE"=10 AND "T"."RESPCODE"=1 AND "TYPE"<>420 AND
          "T"."TIME">=SYSDATE@!-1 AND "REVREQUESTID" IS NULL)
3 - access("T"."PAN"=:B1)
```

**Recommendation:** Add hint /\*+ index (T, IDX\_TLG\_PAN \*/ so index IDX\_TLG\_PAN can be used.

### 5.2.2. SQL ID 17nkd3cxvmr7u - Module JDBC Thin Client

```
select OS_USERNAME, SESSIONID as SID, TIMESTAMP, LOGOFF_TIME, OWNER, OBJ_NAME,
       ACTION, SES_ACTIONS, ACTION_NAME, USERNAME, TERMINAL, COMMENT_TEXT,
       STATEMENTID, ENTRYID, GRANTEE, RETURNCODE, PRIV_USED
from dba_audit_trail
where (TIMESTAMP >= :1 AND LOGOFF_TIME IS NULL)
      OR LOGOFF_TIME > :2
```

## ORDER BY TIMESTAMP, LOGOFF\_TIME

Plan hash value: 2523130228

| Id | Operation             | Name                    | Rows | Bytes | TempSpc | Cost (%CPU) | Time     |
|----|-----------------------|-------------------------|------|-------|---------|-------------|----------|
| 0  | SELECT STATEMENT      |                         |      |       |         | 23773 (100) |          |
| 1  | SORT ORDER BY         |                         | 264K | 44M   | 46M     | 23773 (3)   | 00:04:46 |
| 2  | HASH JOIN RIGHT OUTER |                         | 264K | 44M   |         | 13472 (5)   | 00:02:42 |
| 3  | INDEX FAST FULL SCAN  | I_STMT_AUDIT_OPTION_MAP | 270  | 1080  |         | 2 (0)       | 00:00:01 |
| 4  | HASH JOIN RIGHT OUTER |                         | 264K | 43M   |         | 13468 (5)   | 00:02:42 |
| 5  | INDEX FULL SCAN       | I_SYSTEM_PRIVILEGE_MAP  | 208  | 5200  |         | 1 (0)       | 00:00:01 |
| 6  | HASH JOIN RIGHT OUTER |                         | 264K | 37M   |         | 13465 (5)   | 00:02:42 |
| 7  | INDEX FULL SCAN       | I_SYSTEM_PRIVILEGE_MAP  | 208  | 1040  |         | 1 (0)       | 00:00:01 |
| 8  | HASH JOIN RIGHT OUTER |                         | 264K | 36M   |         | 13463 (5)   | 00:02:42 |
| 9  | INDEX FULL SCAN       | I_AUDIT_ACTIONS         | 181  | 3439  |         | 1 (0)       | 00:00:01 |
| 10 | TABLE ACCESS FULL     | AUD\$                   | 264K | 31M   |         | 13460 (5)   | 00:02:42 |

| Stat Name         | Statement Total | Per Execution | % Snap Total |
|-------------------|-----------------|---------------|--------------|
| Elapsed Time (ms) | 3,394,056       | 6,182.25      | 0.34         |
| CPU Time (ms)     | 2,279,284       | 4,151.70      | 0.25         |
| Executions        | 549             |               |              |
| Buffer Gets       | 35,871,410      | 65,339.54     | 0.02         |
| Disk Reads        | 22,593,919      | 41,154.68     | 63.61        |
| Parse Calls       | 549             | 1.00          | 0.00         |
| Rows              | 26,541,995      | 48,346.07     |              |

The SQL scan AUD\$ table at rate of 8 times per minutes. Currently, AUD\$ size is at 400MB. It'll grow over time.

**Recommendation:** Scan this table at lower rate or maintain AUD\$ to keep its size small.

---

## 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

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

### Optimizing SQL Statements

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

### Tuning Network Resources

|                                 |                                               |
|---------------------------------|-----------------------------------------------|
| <a href="#">Note: 44694.1</a>   | SQL*Net Packet Sizes (SDU and TDU Parameters) |
| <a href="#">Note: 1005123.6</a> | 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 'avg 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.

