

Understanding Performance Tuning in Oracle

Vijay Mahawar

<http://www.mahawar.net/blog>

Saturday, 2 February, 2013

About me



I am Vijay Mahawar, an Oracle Technologist. I am a member of *AIOUG*, *ODTUG* and *OTN*. I am certified in Oracle and hold OCP in Oracle 11g DBA and OCP in Oracle 11g Advanced Developer Track. I am also OPN Certified Specialist.



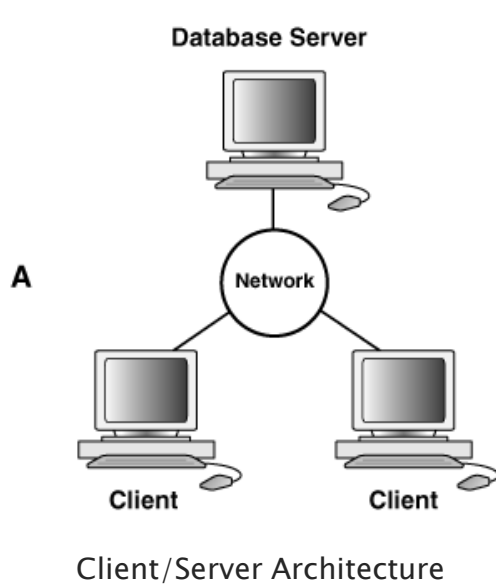
I am associated with Oracle Technologies since year 2002. During this time I have held positions of Trainer, Developer, DBA, Analyst and Consultant in Companies like L&T Infotech Ltd., HCL Technologies Ltd., Dell Services and Sapient Consulting Pvt. Ltd.

What is Performance Tuning?

- ▶ Performance Tuning is optimal utilization of all resources and enabling them to perform to their fullest potential.
- ▶ Performance of the SQL queries of an application often play a big role in the overall performance of the underlying application.
- ▶ The response time may at times be really irritating for the end users if the application doesn't have fine-tuned SQL queries.
- ▶ Differentiate between symptoms and problem. Most tools and techniques give symptoms and that should not be confused with problems.
- ▶ Developers/DBA both have part to play in keeping the database tuned.

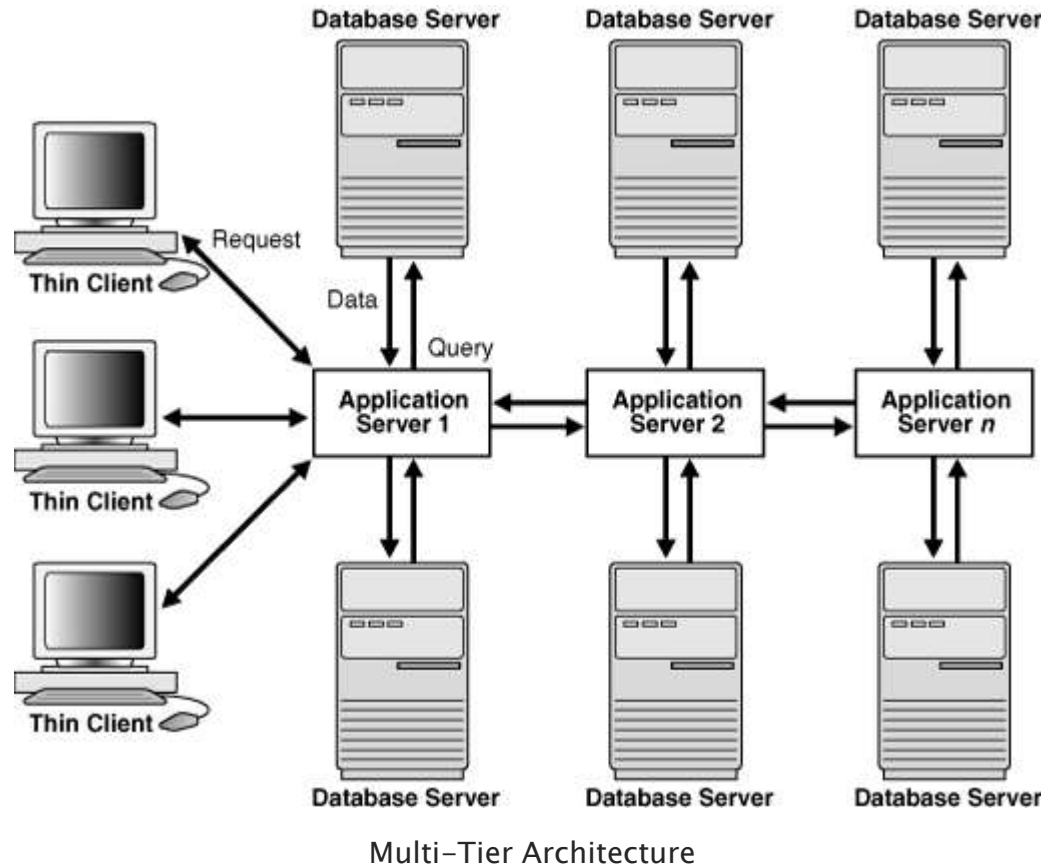


Different layers in application?



Different Layers:

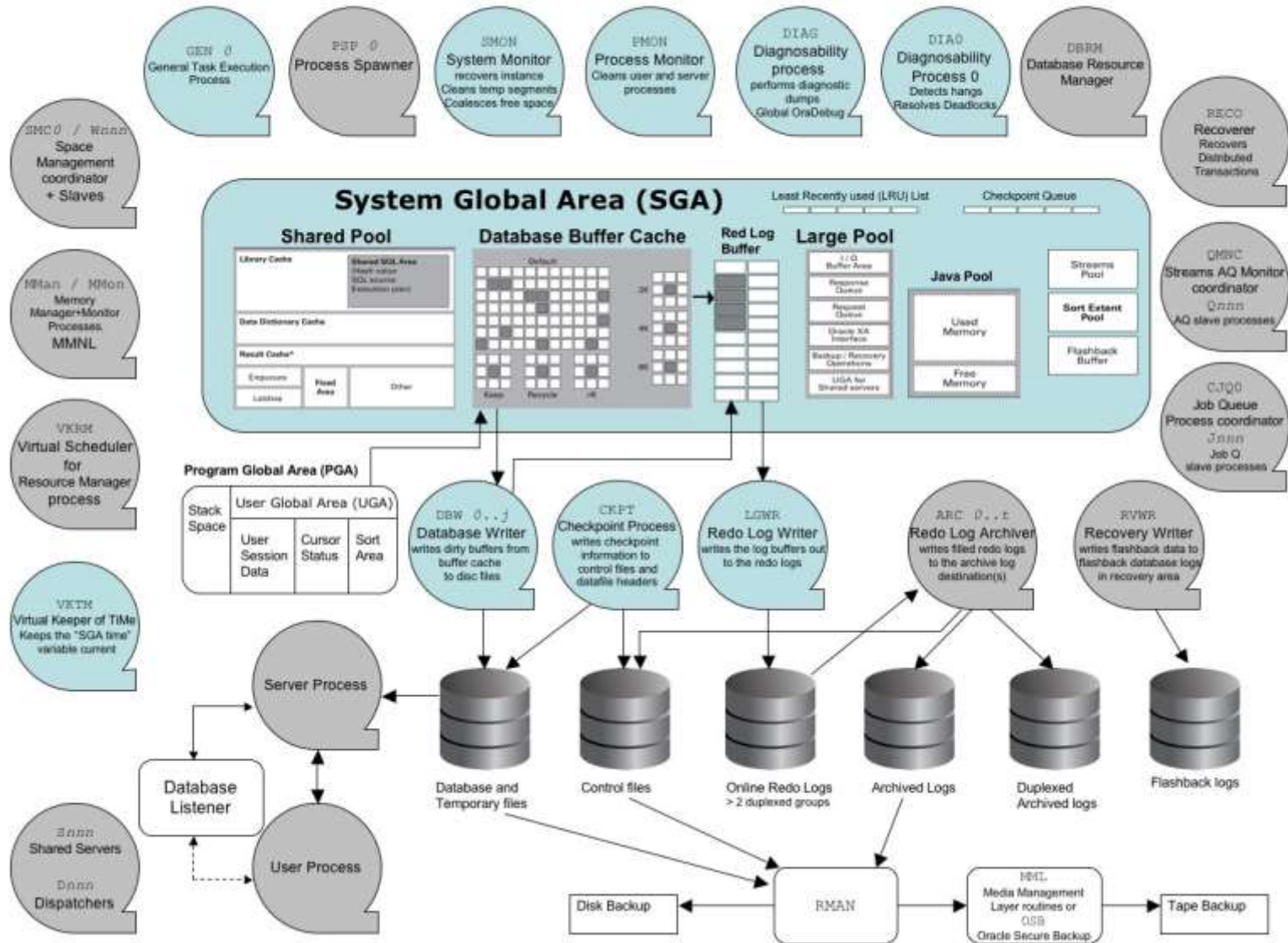
- ▶ Application Interface
- ▶ Network
- ▶ Application Servers
- ▶ Processes
- ▶ Database Instance
- ▶ CPU
- ▶ Disk



Oracle Database 11g Architecture Diagram

Processes in blue are mandatory for the database to be functional

SS64.com



How to identify the tuning area?

	Tuning Areas	Description
P r e c e d e n c e ↓	Database Design (if it's not too late)	Try to normalize to the 3NF. Selective <i>denormalization</i> can provide performance improvements. Always keep the "data access path" in mind. Look at <i>data partitioning, data replication, aggregation tables</i> etc. for <i>DSS – Decision Support System</i>
	Application Tuning	Majority of all Oracle system performance problems are resolved by coding optimal SQL. Scheduling of batch tasks during non-peak hours.
	Memory Tuning	Optimal sizing of database buffers (shared_pool, buffer cache, log buffer, etc) by looking at ADDM report showing <i>wait events, buffer hit ratios, system swapping and paging, etc.</i> Use of <i>pin</i> for large objects into memory to prevent frequent reloads.
	Disk I/O Tuning	Size Database files and properly place them Look for <i>frequent disk sorts, full table scans, missing indexes, row chaining, data fragmentation, etc.</i>
	Eliminate Database Contention	Study database <i>locks, latches and wait events</i> carefully and eliminate where possible.
	Tune the OS	Monitor and tune operating system using OS utilities shown later.

Tuning tools & techniques



Tuning Tools/Programming Techniques	Developer	DBA
Explain Plan (3 ways of using Explain Plan)	✓	●
OEM Console - SQL Monitoring	✓	✓
DBMS_PROFILER/DBMS_HPROF	✓	✓
V\$ Performance Views	✓	✓
Hints/Indexes in SQL Queries (Stats should be updated)	✓	✓
BULK Collect in PL/SQL Programming	✓	●
FORALL in PL/SQL Programming	✓	●
DBMS_PARALLEL_EXECUTE in PL/SQL Programming	✓	✓
RESULT_CACHE in Hint and Functions	✓	●
Netstat (Network)	●	✓
Sar, iostat (Disk)	●	✓
Sar, vmstat (Memory)	●	✓
Sar, vmstat, mpstat, iostat(CPU)	●	✓
SQL_TRACE and TKProf	●	✓
ADDM	●	✓
Statspack (old UTLBSTAT/UTLESTAT)	●	✓
OEM Tuning Pack	●	✓

1. Explain Plan

- ▶ When an SQL statement is passed to the server the Cost Based Optimizer (CBO) uses database statistics to create an execution plan which it uses to navigate through the data.
- ▶ EXPLAIN the statement to check the execution plan that the CBO has created.
- ▶ This will help in diagnosis and often reveal that the query is not using the relevant indexes, or indexes to support the query are missing.
- ▶ Three Methods:
 1. DBMS_XPLAN
 2. EXPLAIN PLAN FOR
 3. SET AUTOTRACE ON for all queries
- ▶ PLAN_TABLE is metadata table for explain plan. utlxplan.sql script creates this table in oracle.



1. Explain Plan – OEM

ORACLE Enterprise Manager 10g Grid Control

Home Targets Deployments Alerts Compliance Jobs Reports

Hosts | Databases | Web Applications | Middleware | Services | Systems | Groups | All Targets

Database Instance dev10em023-d1.oracleads.com > Monitored SQL Executions > Monitored SQL Execution Details

Logged in As SYS

View Report Refresh 5 seconds Stop Refresh

Overview

SQL ID: 33y540fhs1p6
 Parallel: 3
 Execution Started: Sat Nov 21 2009 07:12:24 AM
 Last Refresh Time: Sat Nov 21 2009 07:19:14 AM
 Execution ID: 16777221
 Session: 143
 Fetch Calls: 0

Time: Duration: 6.9m
 Database Time: 16.6m
 PL/SQL & Java: 0.0%

IO & Wait Statistics: IO Count: 260K
 Buffer Gets: 2210
 Wait Activity %: 100

Details

Plan Statistics Parallel Activity

Plan Hash Value: 732509493

Operation	Name	Estimated Rows	Cost	Timeline(411s)	Executions	Actual Rows	Memory	Temp	CPU Activity %	Wait Activity %
SELECT STATEMENT			9510		7					
PX COORDINATOR					7					
PX SEND QC (ORDER)	TQ10003	337P	9510							
SORT ORDER BY		337P	9510							
PX RECEIVE		337P	3260							
PX SEND RANGE	TQ10002	337P	3260		3					
MERGE JOIN CARTESIAN		337P	3260		3					
BUFFER SORT					3		1846K 2211M		18	84
PX RECEIVE		4841M	4678K		3	9682K			40	0.30
PX SEND BROADCAST	TQ10001	4841M	4678K		3	3672K			39	15
MERGE JOIN CARTESIAN		4841M	4678K		3	3224K				
BUFFER SORT					3	15K	27M		0.40	
PX RECEIVE		70K	78		3	209K			0.01	
PX SEND BROADCAST	TQ10000	70K	78		3	209K			0.40	0.15
PX BLOCK ITERATOR		70K	78		3	70K				
TABLE ACCESS FULL	MYONE	70K	78		38	70K				
BUFFER SORT		70K	4678K		156	3224K	9173K		1.81	
PX BLOCK ITERATOR		70K	78		3	70K				

2. Hints

- ▶ A **hint** is an instruction to the optimizer.
- ▶ When writing SQL code, you may know information about the data unknown to the optimizer, usually when statistics are out of date
- ▶ Hints enable you to make decisions normally made by the optimizer, sometimes causing the optimizer to select a plan that it sees as higher cost.

USAGE:

```
/*+ hint */  
/*+ hint(argument) */  
/*+ hint(argument-1 argument-2) */
```

EXAMPLE:

```
select /*+ index(scott.emp ix_emp) */  
from scott.emp emp_alias
```

Hints Categories	Examples
Optimization Approaches and Goals	ALL_ROWS, FIRST_ROWS, CHOOSE
Access Paths	FULL, HASH, INDEX
Query Transformations	FACT, NO_FACT, MERGE, NO_MERGE
Join Orders	LEADING, USE_NL, USE_HASH, USE_MERGE
Parallel Execution	NOPARALLEL, PARALLEL, NOPARALLEL_INDEX, PARALLEL_INDEX
Additional Hints	APPEND, CACHE, DYNAMIC_SAMPLING, RESULT_CACHE, ORDERED

3. BULK Collect and FORALL

- ▶ Without the bulk bind, PL/SQL sends a SQL statement to the SQL engine for each record that is inserted, updated, or deleted leading to context switches that hurt performance.
- ▶ One method of overcoming this performance bottleneck is an Oracle bulk collect.
- ▶ With Oracle bulk collect, the PL/SQL engine tells the SQL engine to collect many rows at once and place them in a collection and switches back to the PL/SQL engine.
- ▶ Syntax:
FETCH BULK COLLECT <cursor_name>
INTO <collection_name>
LIMIT <numeric_expression>;

3. BULK Collect and FORALL

- ▶ The FORALL statement issues a series of static or dynamic DML statements using the collection mostly populated using BULK Collect we saw in previous slide.
- ▶ This will allowing the DML to be run for each row in the collection without requiring a context switch each time.
- ▶ Syntax:
*FORALL <index_name> IN lower_bound..upper_bound
dml_statement SAVE EXCEPTIONS*
or
*FORALL <index_name> IN INDICES OF collection BETWEEN lower_bound and upper_bound
dml_statement SAVE EXCEPTIONS*
- ▶ **SQL%BULK_ROWCOUNT**: cursor attribute gives granular information about the rows affected by each iteration of the FORALL statement
- ▶ **SAVE EXCEPTIONS** and **SQL%BULK_EXCEPTION**: Optional keywords that cause the FORALL loop to continue even if some DML operations fail.

5. V\$ Views – Host System Diagnosis

Tuning Area	V\$ Performance Views	OS Tools
I/O	V\$SYSTEM_EVENT, V\$IOSTAT_CONSUMER_GROUP, V\$IOSTAT_FILE, V\$IOSTAT_FUNCTION, V\$IOSTAT	sar -d (or iostat)
CPU (CPU Time Vs Real Time)	V\$SYSSTAT, V\$SESSTAT, V\$RSRC_CONSUMER_GROUP, V\$SYS_TIME_MODEL	Sar, vmstat, mpstat, iostat
Memory	Initialization parameters: STATISTICS_LEVEL, DB_CACHE_ADVICE, TIMED_STATISTICS or TIMED_OS_STATISTICS	Sar* (vmstat)
Network	V\$IOSTAT_NETWORK	Ping*, traceroute, tracert(windows)

Sar* – System Activity Monitor

Ping* – Packet IntergNet Groper

Tracert* – Trace Route (TTL)

Wait Events

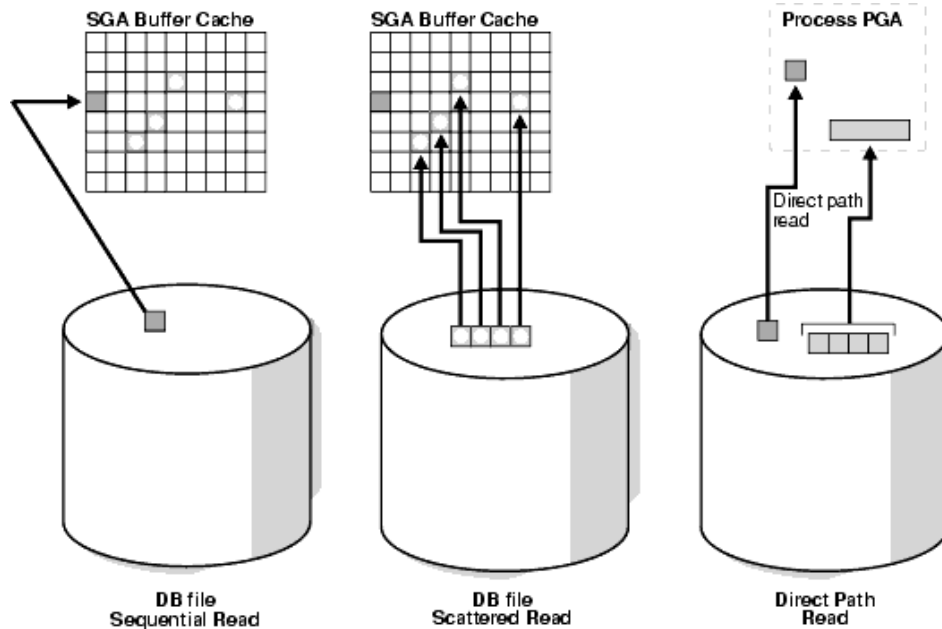


Fig. shows difference between following wait events:

- *db file sequential read* – single block read into one SGA buffer
- *db file scattered read* – multiblock read into many discontinuous SGA buffers
- *direct read* – single or multiblock read into the PGA, bypassing the SGA

On a healthy system, physical read waits should be the biggest waits after the idle waits. However, also consider whether there are db file sequential reads on a large data warehouse that should be seeing mostly full table scans with parallel query.

References & Sources

Type	Link/Author	Reference Comments
Books	<i>Donald K. Burlison (author)</i>	Oracle Tuning: The Definitive Reference
Website	Oracle Documentation	Content and images were referred.
Website	http://ss64.com	for Image of Oracle 11g Architecture Diagram
Website	http://www.orafaq.com	FAQ section for performance tuning was referred.
Website	http://www.oracle-base.com	New features in FORALL in 11g was referred
Website	http://psoug.org	PSOUG.org is an Oracle community resource dedicated to supporting users and developers of the Oracle database.
Webiste	http://www.adp-gmbh.ch	For hints and their classificaitons

Questions?

Thank You



<https://www.mahawar.net/blog>



vmahawar (on twitter)