



Avoiding Common Performance Issues When Scaling RDBMS Applications with Oracle 9i Release 2 and Sun Fire™ Servers Appendices

Glenn P. Fawcett, Strategic Applications Engineering

Sun BluePrints™ OnLine—March 2003



<http://www.sun.com/blueprints>

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95045 U.S.A.
650 960-1300

Part No. 817-2196-10
Revision 1.0, mm/dd/02
Edition: March 2003

Copyright 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries.

This document and the product to which it pertains are distributed under licenses restricting their use, copying, distribution, and decompilation. No part of the product or of this document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, AnswerBook2, docs.sun.com, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and in other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and in other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

U.S. Government Rights—Commercial use. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. a les droits de propriété intellectuels relatants à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et sans la limitation, ces droits de propriété intellectuels peuvent inclure un ou plus des brevets américains énumérés à <http://www.sun.com/patents> et un ou les brevets plus supplémentaires ou les applications de brevet en attente dans les Etats-Unis et dans les autres pays.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a.

Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, AnswerBook2, docs.sun.com, et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciées de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.



Please
Recycle



Adobe PostScript

Avoiding Common Performance Issues When Scaling RDBMS Applications with Oracle 9i Release 2 and Sun Fire™ Servers

Appendices

This document contains appendices to the “Avoiding Common Performance Issues When Scaling RDBMS Applications with Oracle 9i Release 2 and Sun Fire™ Servers” article.

- “Appendix A - Shared Pool Contention Sample `report.txt` Sections” on page 2
- “Appendix B - Example Statspack Report With Some Cache Buffers Chains and Shared Pool Issues.” on page 11
- “Appendix C: `report.txt` to Show Buffer Busy Waits” on page 35
- “Appendix D: Buffer Busy Wait Analysis Scripts.” on page 51

Appendix A - Shared Pool Contention

Sample report.txt Sections

```
>set charwidth 12
>set numwidth 10
>select namespace library,
gets,
round(decode(gethits,0,1,gethits)/decode(gets,0,1,gets),3)
gethitratio,
pins,
round(decode(pinhits,0,1,pinhits)/decode(pins,0,1,pins),3)
pinhitratio,
reloads, invalidations
from stats$lib
```

LIBRARY	GETS	GETHISTRATI	PINS	PINHISTRATI	RELOADS	INVALIDATI
BODY	1537	1	1537	1	0	0
CLUSTER	0	1	0	1	0	0
INDEX	0	1	0	1	0	0
OBJECT	0	1	0	1	0	0
PIPE	0	1	0	1	0	0
SQL AREA	1352453	.727	4697934	.825	68325	414
TABLE/PROCED	2109891	.926	2242030	.923	50004	0
TRIGGER	175304	1	146576	1	0	0

8 rows selected.

```
>set charwidth 27
>set numwidth 12
>select n1.name "Statistic",
n1.change "Total",
round(n1.change/trans.change,2) "Per Transaction",
round(n1.change/logs.change,2) "Per Logon"
from stats$stats n1, stats$stats trans, stats$stats logs
where trans.name='user commits'
and logs.name='logons cumulative'
```

```

and n1.change != 0
order by n1.name

```

Statistic	Total	Per Transact	Per Logon
-----	-----	-----	-----
CPU used by this session	2994273	51.65	1872.59
CPU used when call started	2994995	51.67	1873.04
CR blocks created	25160	.43	15.73
Current blocks converted fo	33	0	.02
DBWR buffers scanned	9119515	157.32	5703.26
DBWR checkpoints	357	.01	.22
DBWR free buffers found	7889475	136.1	4934.01
DBWR lru scans	12652	.22	7.91
DBWR make free requests	11606	.2	7.26
DBWR summed scan depth	9135976	157.6	5713.56
DBWR timeouts	3354	.06	2.1
background checkpoints comp	6	0	0
background checkpoints star	6	0	0
background timeouts	14274	.25	8.93
calls to kcmgas	131970	2.28	82.53
calls to kcmgcs	5588	.1	3.49
calls to kcmgrs	2565404	44.25	1604.38
change write time	10884	.19	6.81
cleanouts and rollbacks - c	21758	.38	13.61
cleanouts only - consistent	112507	1.94	70.36
cluster key scan block gets	110451	1.91	69.08
cluster key scans	60293	1.04	37.71
consistent changes	77404	1.34	48.41
consistent gets	122418713	2111.8	76559.55
cursor authentications	232266	4.01	145.26
data blocks consistent read	66498	1.15	41.59
db block changes	2484777	42.86	1553.96
db block gets	5052626	87.16	3159.87
deferred (CURRENT) block cl	69510	1.2	43.47

dirty buffers inspected	82	0	.05
enqueue conversions	1497	.03	.94
enqueue releases	376376	6.49	235.38
enqueue requests	376384	6.49	235.39
enqueue timeouts	1	0	0
enqueue waits	374	.01	.23
exchange deadlocks	15	0	.01
execute count	2028794	35	1268.79
free buffer inspected	41877	.72	26.19
free buffer requested	6244790	107.73	3905.43
immediate (CR) block cleano	134267	2.32	83.97
immediate (CURRENT) block c	76661	1.32	47.94
logons cumulative	1599	.03	1
logons current	-75	0	-.05
messages received	101263	1.75	63.33
messages sent	101263	1.75	63.33
no work - consistent read g	84178656	1452.13	52644.56
opened cursors cumulative	719512	12.41	449.98
opened cursors current	-940	-.02	-.59
parse count	1352172	23.33	845.64
parse time cpu	1608857	27.75	1006.16
parse time elapsed	24331159	419.73	15216.48
physical reads	6173978	106.5	3861.15
physical writes	146954	2.54	91.9
process last non-idle time	-1287945463	-22217.83	-805469.33
recursive calls	7166506	123.63	4481.87
recursive cpu usage	444235	7.66	277.82
redo blocks written	623622	10.76	390.01
redo buffer allocation retr	6	0	0
redo entries	1344467	23.19	840.82
redo entries linearized	1344467	23.19	840.82
redo log space requests	17	0	.01
redo log space wait time	204	0	.13
redo size	282849560	4879.32	176891.53
redo small copies	1059380	18.27	662.53
redo synch time	51128	.88	31.97

redo synch writes	73453	1.27	45.94
redo wastage	25276367	436.03	15807.61
redo write time	49588	.86	31.01
redo writer latching time	359	.01	.22
redo writes	95232	1.64	59.56
rollback changes - undo rec	9830	.17	6.15
rollbacks only - consistent	3381	.06	2.11
session connect time	2558135751	44129.38	1599834.74
session logical reads	127366348	2197.15	79653.75
session pga memory	574025188	9902.28	358990.11
session pga memory max	615821780	10623.29	385129.32
session uga memory	15076776	260.08	9428.88
session uga memory max	205334004	3542.13	128414.01
sorts (disk)	128	0	.08
sorts (memory)	119745	2.07	74.89
sorts (rows)	9656777	166.59	6039.26
summed dirty queue length	1616	.03	1.01
table fetch by rowid	38904658	671.13	24330.62
table fetch continued row	753465	13	471.21
table scan blocks gotten	7512582	129.6	4698.3
table scan rows gotten	247732036	4273.53	154929.35
table scans (long tables)	186	0	.12
table scans (short tables)	690166	11.91	431.62
transaction rollbacks	62	0	.04
transaction tables consiste	34	0	.02
transaction tables consiste	10717	.18	6.7
user calls	2457015	42.38	1536.59
user commits	57969	1	36.25
user rollbacks	2187	.04	1.37
write requests	8330	.14	5.21

95 rows selected.

>set charwidth 27

>set numwidth 12

```
>select nl.event "Event Name",
nl.event_count "Count",
nl.time_waited "Total Time",
```

```
(nl.time_waited/nl.event_count) "Average Time"
```

```
from stats$event nl
```

```
where nl.event_count > 0
```

```
order by nl.time_waited desc
```

Event Name	Count	Total Time	Average Time
client message	2463397	461123489	187.19008304
latch free	13941018	24625669	1.7664182773
rdbms ipc message	108655	6855524	63.094418112
smon timer	50	1440151	28803.02
pmon timer	11935	1427491	119.60544617
library cache pin	8587	1105899	128.78758589
row cache lock	6099	797871	130.81997049
library cache load lock	1146	248904	217.19371728
db file sequential read	2292441	179507	.07830386911
db file scattered read	692573	107550	.15529048923
enqueue	502	51896	103.37848606
log file sync	70982	51115	.72011214111
log file parallel write	95222	49540	.52025792359
db file parallel write	8329	21877	2.626605835
rdbms ipc reply	352	4507	12.803977273
buffer busy waits	27469	1975	.07189923186
write complete waits	276	1394	5.0507246377
control file parallel write	894	371	.41498881432
log file space/switch	17	204	12
db file single write	636	157	.24685534591
buffer deadlock	15	30	2
control file sequential rea	479	20	.04175365344
log file sequential read	6	1	.16666666667
log file single write	12	0	0

24 rows selected.

```
>set charwidth 18
```

```
>set numwidth 11
```

```
>select name latch_name, gets, misses,
```

```
round(decode(gets-misses,0,1,gets-misses)/decode(gets,0,1,gets),3)
```

```

hit_ratio,
sleeps,
round(sleeps/decode(misses,0,1,misses),3) "SLEEPS/MISS"
from stats$latches
where gets != 0
order by name

```

LATCH_NAME	GETS	MISSES	HIT_RATIO	SLEEPS	SLEEPS/MISS
archive control	6	0	1	0	0
cache buffer handl	424437	96	1	26	.271
cache buffers chai	240882509	89405	1	16508	.185
cache buffers lru	6262012	267660	.957	62627	.234
dml lock allocatio	427618	151	1	69	.457
enqueuees	1074219	2108	.998	1007	.478
latch wait list	19894023	530221	.973	95671	.18
library cache	16576631	2774984	.833	4361706	1.572
library cache load	330908	328	.999	144	.439
library cache pin	19461210	1539713	.921	1272391	.826
messages	406063	881	.998	391	.444
multiblock read ob	1756087	257	1	10	.039
process allocation	1402	0	1	0	0
redo allocation	1690286	6241	.996	2870	.46
redo copy	25	10	.6	1	.1
row cache objects	25597969	1342118	.948	2286145	1.703
sequence cache	277967	59	1	17	.288
session allocation	1498147	2200	.999	1011	.46
session idle bit	5200481	1835	1	31	.017
session switching	3935	0	1	0	0
shared pool	19632472	4577055	.767	5840203	1.276
system commit numb	4450111	2607	.999	60	.023
transaction alloca	267729	104	1	47	.452
undo global data	971347	742	.999	91	.123
user lock	6025	9	.999	30	3.333

```

25 rows selected.
>set numwidth 18
>select name latch_name,

```

```

immed_gets nowait_gets,
immed_miss nowait_misses,
round(decode(immed_gets-immed_miss,0,1,immed_gets-immed_miss)/
decode(immed_gets,0,1,immed_gets),
3)

```

```

nowait_hit_ratio
from stats$latches
where immed_gets != 0
order by name

```

LATCH_NAME	NOWAIT_GETS	NOWAIT_MISSES	NOWAIT_HIT_RATIO
cache buffers chain	13505794	2905	1
cache buffers lru	118723311	5656720	.952
library cache	1347183	207441	.846
library cache pin	278756	38909	.86
redo copy	3332148	26	1
row cache objects	187581	63229	.663
session idle bit	33	0	1

7 rows selected.

```
>set charwidth 39
```

```
>select name, value from v$parameter where isdefault = 'FALSE' order by name
```

NAME	VALUE
audit_trail	NONE
background_dump_dest	/u00/home/oracle/admin/mctp/bdump
cache_size_threshold	3840
checkpoint_process	TRUE
compatible	7.1.6.0
control_files	/a06/oradata/mctp/control01.ctl, /a07/o
core_dump_dest	/u00/home/oracle/admin/mctp/cdump
db_block_buffers	38400
db_block_size	8192
db_file_multiblock_read_count	16
db_files	400
db_name	mctp
distributed_transactions	243

dml_locks	2500
enqueue_resources	3000
gc_db_locks	38400
ifile	/u00/home/oracle/admin/mctp/pfile/confi
log_archive_dest	/archive/mctp/mctp_
log_archive_format	%S.arc
log_archive_start	TRUE
log_buffer	1024000
log_checkpoint_interval	100000000
log_checkpoints_to_alert	TRUE
log_simultaneous_copies	32
log_small_entry_max_size	200
max_dump_file_size	61440
mts_max_dispatchers	0
mts_max_servers	0
mts_servers	0
mts_service	mctp
open_cursors	250
optimizer_mode	RULE
processes	800
remote_login_passwordfile	NONE
resource_limit	TRUE
rollback_segments	r01, r02, r03, r04, r05, r06, r07, r08,
sequence_cache_entries	1000
sequence_cache_hash_buckets	23
sessions	885
shared_pool_reserved_min_alloc	8192
shared_pool_reserved_size	104857600
shared_pool_size	104857600
sort_area_retained_size	102400
sort_area_size	2048000
temporary_table_locks	885
timed_statistics	TRUE
transactions	973
transactions_per_rollback_segment	71
use_ism	FALSE

```

user_dump_dest                /u00/home/oracle/admin/mctp/udump
50 rows selected.
>set charwidth 15
>set numwidth 8
>select * from stats$dc
where get_reqs != 0 or scan_reqs != 0 or mod_reqs != 0
NAME                          GET_REQS GET_MISS SCAN_REQ SCAN_MIS MOD_REQS COUNT      CUR_USAG
-----
dc_tablespaces                10402      2      0      0      0      15      14
dc_free_extents                1122      346     346      0     691      24      22
dc_segments                   605612    1497      0      0     553     133     132
dc_rollback_seg               2325      0      0      0      0      25      24
dc_used_extents                687      344      0      0     687      17      15
dc_tablespace_q                2         1      0      0      2       3       2
dc_users                      156620    784      0      0     12      67      65
dc_user_grants                14949     742      0      0      0      59      58
dc_objects                    1995457   92383      0      0      0    11036   11037
dc_tables                     1471823   739      0      0      0      210     209
dc_columns                    31863426  53112   718239   30534      0     2504     2501
dc_table_grants               718133    1957      0      0      0      591     589
dc_indexes                    4660215   1478   608321    604      0      414     413
dc_constraint_d                382      46   85797     91      0      19      16
dc_constraint_d                0         0   56735     43      0       1       0
dc_synonyms                   524451    1899      0      0      0     213     212
dc_usernames                  681225    2801      0      0      2     118     117
dc_object_ids                  1         1      0      0      0       3       2
dc_sequences                  102827     96      0      0   71190      29      28
dc_sequence_gra               34248     281      0      0      0     119      68
dc_tablespaces                 346      0      0      0     346      5       4
dc_profiles                   2246      0      0      0      0      17      3
22 rows selected.
>set charwidth 20
>select * from stats$dates
STATS_GATHER_TIMES
-----
23-feb-98 13:00:32

```

23-feb-98 16:59:15

2 rows selected.

>spool off

Appendix B - Example Statspack Report With Some Cache Buffers Chains and Shared Pool Issues.

STATSPACK report for

DB Name	DB Id	Instance	Inst Num	Release	OPS	Host
PRD	1553049205	PRD	1	8.1.7.0.0	NO	bao-01

	Snap Id	Snap Time	Sessions
Begin Snap:	1	05-Apr-01 17:02:02	22,655
End Snap:	2	05-Apr-01 17:22:57	22,655
Elapsed:		20.92 (mins)	

Cache Sizes

~~~~~

|                   |        |                   |            |
|-------------------|--------|-------------------|------------|
| db_block_buffers: | 131072 | log_buffer:       | 2097152    |
| db_block_size:    | 8192   | shared_pool_size: | 1610612736 |

Load Profile

~~~~~

	Per Second	Per Transaction
Redo size:	665,483.69	12,405.97
Logical reads:	400,554.45	7,467.15
Block changes:	3,904.11	72.78
Physical reads:	194.03	3.62
Physical writes:	133.91	2.50

User calls:	7,104.51	132.44
Parses:	2,967.97	55.33
Hard parses:	9.62	0.18
Sorts:	1,345.80	25.09
Logons:	3.35	0.06
Executes:	12,724.95	237.22
Transactions:	53.64	

% Blocks changed per Read:	0.97	Recursive Call %:	74.21
Rollback per transaction %:	30.44	Rows per Sort:	37.34

Instance Efficiency Percentages (Target 100%)

```

~~~~~
Buffer Nowait %: 100.00      Redo NoWait %: 100.00
Buffer Hit %: 99.95        In-memory Sort %: 100.00
Library Hit %: 100.07      Soft Parse %: 99.68
Execute to Parse %: 76.68   Latch Hit %: 97.37
Parse CPU to Parse Elapsed %: 88.36   % Non-Parse CPU: 99.98

```

```

Shared Pool Statistics      Begin   End
-----  -----
Memory Usage %: 72.00      73.22
% SQL with executions>1: 61.70      60.38
% Memory for SQL w/exec>1: 52.68      53.91

```

Top 5 Wait Events

```

~~~~~
Event                               Wait   % Total
                                Waits Time(cs) Wt Time
-----  -----  -----  -----
db file sequential read             224,834 183,231 42.01
log file sync                       55,993 104,683 24.00
latch free                          43,475  98,378 22.55
db file parallel write              31,547  12,797  2.93
buffer busy waits                   10,949  11,319  2.59
-----  -----  -----  -----

```

Wait Events for DB: PRD Instance: PRD Snaps: 1 -2

-> cs - centisecond - 100th of a second

-> ms - millisecond - 1000th of a second

-> ordered by wait time desc, waits desc (idle events last)

Event	Waits	Timeouts	Total Wait Time (cs)	Avg wait (ms)	Waits /txn
db file sequential read	224,834	0	183,231	8	3.3
log file sync	55,993	1	104,683	19	0.8
latch free	43,475	32,529	98,378	23	0.6
db file parallel write	31,547	0	12,797	4	0.5
buffer busy waits	10,949	26	11,319	10	0.2
log file parallel write	42,436	0	8,281	2	0.6
file open	4,025	0	6,216	15	0.1
library cache pin	217	10	5,788	267	0.0
enqueue	926	0	1,484	16	0.0
SQL*Net break/reset to clien	11,118	0	1,389	1	0.2
direct path write	1,183	0	512	4	0.0
SQL*Net more data to client	56,328	0	508	0	0.8
db file scattered read	321	0	370	12	0.0
control file parallel write	398	0	356	9	0.0
direct path read	706	0	308	4	0.0
LGWR wait for redo copy	2,670	15	251	1	0.0
row cache lock	17	0	14	8	0.0
buffer deadlock	101	101	10	1	0.0
control file sequential read	31	0	7	2	0.0
refresh controlfile command	6	0	5	8	0.0
SQL*Net message from client	9,090,423	0	#####	1262	135.0
SQL*Net more data from clien	58,204	0	891,723	153	0.9
SQL*Net message to client	9,090,199	0	4,769	0	135.0

Background Wait Events for DB: PRD Instance: PRD Snaps: 1 -2

-> ordered by wait time desc, waits desc (idle events last)

Event	Waits	Timeouts	Total Wait Time (cs)	Avg wait (ms)	Waits /txn
db file parallel write	31,574	0	12,810	4	0.5
log file parallel write	42,524	0	8,289	2	0.6
control file parallel write	397	0	354	9	0.0
LGWR wait for redo copy	2,676	15	252	1	0.0
db file sequential read	101	0	58	6	0.0
db file scattered read	20	0	16	8	0.0
latch free	2	2	2	10	0.0
control file sequential read	12	0	1	1	0.0
buffer busy waits	1	0	0	0	0.0
rdbms ipc message	136,541	804	335,275	25	2.0
pmon timer	402	394	124,914	3107	0.0
smon timer	6	5	124,089	#####	0.0

SQL ordered by Gets for DB: PRD Instance: PRD Snaps: 1 -2

-> End Buffer Gets Threshold: 10000

-> Note that resources reported for PL/SQL includes the resources used by all SQL statements called within the PL/SQL code. As individual SQL statements are also reported, it is possible and valid for the summed total % to exceed 100

Buffer Gets	Executions	Gets per Exec	% Total	Hash Value
11,397,700	861	13,237.7	2.3	2182723903
SELECT SYSDATE FROM SYS.DUAL				
9,880,093	30,476	324.2	2.0	1681749616
begin FND_FLEX_WORKFLOW_APIS.VALIDATE_COMBINATION (:v1, :v2, :v3, :v4, :v5); end;				
8,305,470	5,305	1,565.6	1.7	2054874295

```

select O.PROFILE_OPTION_NAME, V.PROFILE_OPTION_VALUE,          V.L
EVEL_ID from      FND_PROFILE_OPTIONS O, FND_PROFILE_OPTION_VALUES
V where  O.START_DATE_ACTIVE <= SYSDATE and      (nvl(O.END_DATE_A
CTIVE,SYSDATE) >= SYSDATE) and      ((V.LEVEL_ID = 10001 and V.LEV
EL_VALUE = 0)   or      (V.LEVEL_ID = 10002 and V.LEVEL_VALUE = :a

```

```

7,828,374          586          13,359.0      1.6      1318850758
SELECT DISTINCT P.USER_CONCURRENT_PROGRAM_NAME, A.APPLICATION_ID
, A.APPLICATION_NAME, A.APPLICATION_SHORT_NAME, P.CONCURRENT_PRO
GRAM_NAME, P.CONCURRENT_PROGRAM_ID, P.APPLICATION_ID PROGRAM_APP
LICATION_ID, '$SRS$. ' || P.CONCURRENT_PROGRAM_NAME DESCRIPTIVE_FLE
XFIELD_NAME, P.SRS_FLAG, P.PRINTER_NAME, P.OUTPUT_PRINT_STYLE, P

```

```

4,371,997          10,064          434.4      0.9      161371248
BEGIN          ARP_PROCESS_TAX.interface_tax(
p_sql => :l_sqlstmt,          p_bind1 => :l_bind1:il
_bind1,          p_bind2 => :l_bind2:il_bind2,
          p_bind3 => :l_bind3:il_bind3,          p_bind
4 => :l_bind4:il_bind4,          p_new_tax_amount =>:l_

```

```

3,462,984          28,508          121.5      0.7      1052917712
SELECT V.PROFILE_OPTION_VALUE,V.LEVEL_ID FROM FND_PROFILE_OPTI
ONS O,FND_PROFILE_OPTION_VALUES V WHERE O.PROFILE_OPTION_NAME =
:b1 AND O.START_DATE_ACTIVE <= SYSDATE AND (NVL(O.END_DATE_AC
TIVE,SYSDATE) >= SYSDATE ) AND (V.LEVEL_ID = 10001 AND V.LEVEL_
VALUE = 0 ) AND O.PROFILE_OPTION_ID = V.PROFILE_OPTION_ID AND O

```

```

3,351,487          3      1,117,162.3      0.7      2448097883
SELECT decode ( : P_BREAK_ID , 1 , nvl ( msi.SEGMENT1 , ' ' ) ,
2 , nvl ( to_number ( to_char ( mmt . transaction_date , 'J' )
) , 0 ) , 3 , nvl ( mtxt1.transaction_type_name , ' ' ) , 4 ,
nvl ( mtxt1.transaction_type_name , ' ' ) , 5 , nvl ( mtr .
reason_name , ' ' ) , 6 , nvl ( mmt . subinventory_code , ' ' )

```

```

3,101,767          3      1,033,922.3      0.6      654568447
SELECT decode ( : P_BREAK_ID , 1 , nvl ( msi.SEGMENT1 , ' ' ) ,

```

```

2 , nvl ( to_number ( to_char ( mmt . transaction_date , 'J' )
) , 0 ) , 3 , nvl ( mtxt1.transaction_type_name , ' ' ) , 4 ,
nvl ( mtxt1.transaction_type_name , ' ' ) , 5 , nvl ( mtr .
reason_name , ' ' ) , 6 , nvl ( mmt . subinventory_code , ' ' )

```

```

3,008,570      30,474      98.7      0.6      548943000
SELECT 'NON', 'NON' , 0 , A.SUBINVENTORY , 0 , A.LOCATOR_ID , SU

```

SQL ordered by Gets for DB: PRD Instance: PRD Snaps: 1 -2

-> End Buffer Gets Threshold: 10000

-> Note that resources reported for PL/SQL includes the resources used by all SQL statements called within the PL/SQL code. As individual SQL statements are also reported, it is possible and valid for the summed total % to exceed 100

SQL ordered by Reads for DB: PRD Instance: PRD Snaps: 1 -2

-> End Disk Reads Threshold: 1000

```

Physical Reads  Executions  Reads per Exec % Total  Hash Value
-----
13,448          11,733          1.1      5.5      598666692
insert into MTL_DEMAND(DEMAND_ID,ORGANIZATION_ID,INVENTORY_ITEM_
ID,DEMAND_SOURCE_TYPE,DEMAND_SOURCE_HEADER_ID,PRIMARY_UOM_QUANTI
TY,UOM_CODE,COMPLETED_QUANTITY,REQUIREMENT_DATE,RESERVATION_TYPE
, LAST_UPDATE_DATE, LAST_UPDATED_BY, CREATION_DATE, CREATED_BY, DEMAN
D_SOURCE_LINE, DEMAND_SOURCE_DELIVERY, USER_LINE_NUM, USER_DELIVERY

```

```

7,973          30          265.8      3.3      472404750
SELECT ROW_ID,ACCRUAL_POSTED_FLAG,ACCTS_PAY_CODE_COMBINATION_ID,
ASSETS_ADDITION_FLAG,ATTRIBUTE1,ATTRIBUTE10,ATTRIBUTE11,ATTRIBUT
E12,ATTRIBUTE13,ATTRIBUTE14,ATTRIBUTE15,ATTRIBUTE2,ATTRIBUTE3,AT
TRIBUTE4,ATTRIBUTE5,ATTRIBUTE6,ATTRIBUTE7,ATTRIBUTE8,ATTRIBUTE9,
ATTRIBUTE_CATEGORY,AWT_FLAG,AWT_GROSS_AMOUNT,AWT_GROUP_ID,AWT_IN

```

```

7,747      10,064      0.8      3.2      161371248
BEGIN      ARP_PROCESS_TAX.interface_tax(
p_sql => :l_sqlstmt,      p_bind1 => :l_bind1:il
_bind1,      p_bind2 => :l_bind2:il_bind2,
p_bind3 => :l_bind3:il_bind3,      p_bind
4 => :l_bind4:il_bind4,      p_new_tax_amount =>:l_

6,222      80,080      0.1      2.6      3775588697
SELECT TAX.CUSTOMER_TRX_LINE_ID,TAX.CUSTOMER_TRX_ID FROM RA_CU
STOMER_TRX_LINES TAX,RA_CUSTOMER_TRX_LINES LINE,AR_VAT_TAX V1,AR
_VAT_TAX V2 WHERE TAX.LINK_TO_CUST_TRX_LINE_ID = LINE.PREVIOUS_
CUSTOMER_TRX_LINE_ID AND LINE.CUSTOMER_TRX_LINE_ID = :b1 AND T
AX.LINE_NUMBER = :b2 AND TAX.VAT_TAX_ID = V1.VAT_TAX_ID AND V1

5,887      15,388      0.4      2.4      2254232812
select nvl(ITEM_COST,0) ,COST_UPDATE_ID into :b0,:b1 from CST_I
TEM_COSTS where ((INVENTORY_ITEM_ID=:b2 and ORGANIZATION_ID=:b3)
and COST_TYPE_ID=:b4)

5,641      11,732      0.5      2.3      2890535394
insert into MTL_DEMAND_INTERFACE(SCHEDULE_GROUP_ID,ATP_GROUP_ID,
AUTODETAIL_GROUP_ID,SESSION_ID,DEMAND_TYPE,DEMAND_SOURCE_TYPE,DE
MAND_SOURCE_HEADER_ID,DEMAND_SOURCE_LINE,DEMAND_SOURCE_DELIVERY,
LOCK_FLAG,ATP_CHECK,ATP_COMPONENTS_FLAG,ACTION_CODE,VALIDATE_ROW
S,TRANSACTION_MODE,PROCESS_FLAG,PARTIALS_ALLOWED_FLAG,DETAIL_RES

5,007      11,732      0.4      2.1      1488580214
SELECT DEMAND_ID,PRIMARY_UOM_QUANTITY,RESERVATION_QUANTITY, LINE
_ITEM_QUANTITY,LINE_ITEM_RESERVATION_QTY,COMPLETED_QUANTITY, NVL
(EXPLOSION_GROUP_ID,0) ,ROWIDTOCHAR(ROWID) ,UOM_CODE,RTO_MODEL_SOU
RCE_LINE,DEMAND_TYPE FROM MTL_DEMAND WHERE DEMAND_SOURCE_TYPE =
:dsrc_type AND DEMAND_SOURCE_HEADER_ID = :header_id AND DEMAND_

4,541      1,491      3.0      1.9      1023964613
SELECT INVOICE_TYPE,VENDOR_NAME,VENDOR_NUMBER,VENDOR_SITE_CODE,I
NVOICE_DATE,INVOICE_NUM,INVOICE_CURRENCY_CODE,INVOICE_AMOUNT,PAY

```

MENT_CURRENCY_CODE, PAYMENT_CROSS_RATE_DATE, PAYMENT_CROSS_RATE_TY
PE, PAY_CURR_INVOICE_AMOUNT, PAYMENT_CROSS_RATE, FREIGHT_AMOUNT, DES
CRIPTION, USSGL_TRANSACTION_CODE, PROJECT, TASK, EXPENDITURE_TYPE, EX

4,253 3,491 1.2 1.7 3386498401

SQL ordered by Reads for DB: PRD Instance: PRD Snaps: 1 -2

-> End Disk Reads Threshold: 1000

SQL ordered by Executions for DB: PRD Instance: PRD Snaps: 1 -2

-> End Executions Threshold: 100

Executions	Rows Processed	Rows per Exec	Hash Value
182,883	0	0.0	3108566298
SELECT SUBSTR(DESCRIPTION,1,50) GLOBAL_DESCRIPTION FROM FND_DESC R_FLEX_CONTEXTS_VL WHERE DESCRIPTIVE_FLEXFIELD_NAME = '' AND SUB STR(DESCRIPTIVE_FLEX_CONTEXT_CODE, 4, 2) = 'US' AND ROWNUM < 2 A ND SUBSTR(DESCRIPTIVE_FLEX_CONTEXT_CODE, 7, 8) = 'ARXTWMAI' AND ENABLED_FLAG = 'Y'			
80,080	0	0.0	3710692446
SELECT TAX.CUSTOMER_TRX_LINE_ID, TAX.CUSTOMER_TRX_ID FROM RA_CU STOMER_TRX_LINES TAX, RA_CUSTOMER_TRX_LINES LINE, AR_VAT_TAX V1, AR _VAT_TAX V2 WHERE TAX.LINK_TO_CUST_TRX_LINE_ID = LINE.PREVIOUS_ CUSTOMER_TRX_LINE_ID AND LINE.CUSTOMER_TRX_LINE_ID = :b1 AND T AX.VAT_TAX_ID = V1.VAT_TAX_ID AND V1.TAX_CODE = V2.TAX_CODE AN			
80,080	0	0.0	3775588697
SELECT TAX.CUSTOMER_TRX_LINE_ID, TAX.CUSTOMER_TRX_ID FROM RA_CU STOMER_TRX_LINES TAX, RA_CUSTOMER_TRX_LINES LINE, AR_VAT_TAX V1, AR _VAT_TAX V2 WHERE TAX.LINK_TO_CUST_TRX_LINE_ID = LINE.PREVIOUS_ CUSTOMER_TRX_LINE_ID AND LINE.CUSTOMER_TRX_LINE_ID = :b1 AND T AX.LINE_NUMBER = :b2 AND TAX.VAT_TAX_ID = V1.VAT_TAX_ID AND V1			

```

        66,200          66,200          1.0    792077673
begin ARP_STAX_204.find_tax_exemption_id(:bill_cust,:ship_cust,:
ship_site,:tax_c,:item_id,:trx_d,:exempt_flag,:insrt,:reason,:ce
rt,:pct_exempt,:ins_flag,:exempt_id,:exempt_type); end;

```

```

        66,161          66,161          1.0    4261887034
BEGIN      APPS.arp_stax.find_tax_exemption_id( bill_to_custome
r_id=>:bill_to_cust_id,  ship_to_customer_id=>:ship_to_cust_id,
  ship_to_site_id=>:site_use_id,  tax_code=>:tax_code,  invent
ory_item_id=>:inventory_item_id,  trx_date=>:trx_date,  tax_ex
empt_flag=>:tax_exempt_flag,  reason_code=>:xmpt_reason,  certi

```

```

        58,482          0          0.0    1181450487
select '1' into :b0 from SO_LINES where ((HEADER_ID=:b1 and REV
ENUE_AMOUNT is not null ) and rownum =1)

```

```

        58,449          0          0.0    3215007305
select '1' into :b0 from SO_HEADERS where (HEADER_ID=:b1 and NV
L(OPEN_FLAG,'N')='N')

```

```

        50,666          0          0.0    2273107589
SELECT SUBSTR(DESCRIPTION,1,50) GLOBAL_DESCRIPTION FROM FND_DESC
R_FLEX_CONTEXTS_VL WHERE DESCRIPTIVE_FLEXFIELD_NAME = 'JG_SO_LIN
E_ATTRIBUTES' AND SUBSTR(DESCRIPTIVE_FLEX_CONTEXT_CODE, 4, 2) =
'US' AND ROWNUM < 2 AND SUBSTR(DESCRIPTIVE_FLEX_CONTEXT_CODE, 7,
8) = 'OEXOEMOE' AND ENABLED_FLAG = 'Y'

```

```

SQL ordered by Sharable Memory for DB: PRD Instance: PRD Snaps: 1 -2
-> End Sharable Memory Threshold: 1048576

```

```

Sharable Mem (b)  Executions  % Total  Hash Value
-----

```

2,633,778 977 0.1 1097169786

```
SELECT ORDER_NUMBER,LINE_NUM,SCHEDULE_DATE,DATE_REQUESTED,WIP_JOB_NUMBER,LOAD_SEQ_NUMBER,PLAN_PROD_SEQ,CUSTOMER_ITEM_NUMBER,CUSTOMER_ITEM_DESC,SERIAL_NUMBER,REVISION,SUBINVENTORY,LOT_NUMBER,INVENTORY_LOCATION_ID,ORGANIZATION_CODE,ORDER_REQUESTED_QUANTITY,ORDER_CANCELLED_QUANTITY,LINE_SHIPPED_QUANTITY,UNIT_CODE,CONTAINER
```

1,373,328 7,488 0.1 2869084281

```
select * from TAX_LINES_CREATE_V where trx_line_id = :b1 order by trx_header_id, trx_line_id, nvl(amount_includes_tax_flag, 'N') desc, compounding_precedence, tax_line_number
```

1,235,352 10,062 0.1 3374439543

```
SELECT * from SO_TAX_LINES_SUMMARY_V WHERE trx_header_id = :b1 ORDER BY trx_header_id, trx_line_id, location_qualifier
```

Instance Activity Stats for DB: PRD Instance: PRD Snaps: 1 -2

Statistic	Total	per Second	per Trans
CPU used by this session	2,239,486	1,784.5	33.3
CPU used when call started	2,238,704	1,783.8	33.3
CR blocks created	814,403	648.9	12.1
DBWR buffers scanned	246,738	196.6	3.7
DBWR checkpoint buffers written	18,913	15.1	0.3
DBWR free buffers found	161,600	128.8	2.4
DBWR lru scans	10,942	8.7	0.2
DBWR make free requests	48,243	38.4	0.7
DBWR summed scan depth	246,738	196.6	3.7
DBWR transaction table writes	288	0.2	0.0
DBWR undo block writes	44,705	35.6	0.7
SQL*Net roundtrips to/from client	9,162,976	7,301.2	136.1
background timeouts	1,239	1.0	0.0

branch node splits	21	0.0	0.0
buffer is not pinned count	321,548,556	256,214.0	4,776.4
buffer is pinned count	264,890,861	211,068.4	3,934.7
bytes received via SQL*Net from c	2,300,424,099	1,833,007.3	34,171.0
bytes sent via SQL*Net to client	1,003,374,066	799,501.3	14,904.3
calls to get snapshot scn: kcmgss	14,339,994	11,426.3	213.0
calls to kcmgas	65,363	52.1	1.0
calls to kcmgcs	41,888	33.4	0.6
change write time	15,711	12.5	0.2
cleanouts and rollbacks - consist	515,283	410.6	7.7
cleanouts only - consistent read	16,725	13.3	0.3
cluster key scan block gets	79,993	63.7	1.2
cluster key scans	10,949	8.7	0.2
commit cleanout failures: block l	19,547	15.6	0.3
commit cleanout failures: buffer	3	0.0	0.0
commit cleanout failures: callbac	259	0.2	0.0
commit cleanout failures: cannot	884	0.7	0.0
commit cleanouts	776,033	618.4	11.5
commit cleanouts successfully com	755,303	601.8	11.2
consistent changes	44,839,442	35,728.6	666.1
consistent gets	483,112,502	384,950.2	7,176.3
cursor authentications	161	0.1	0.0
data blocks consistent reads - un	44,848,718	35,736.0	666.2
db block changes	4,899,657	3,904.1	72.8
db block gets	18,483,962	14,728.3	274.6
deferred (CURRENT) block cleanout	170,559	135.9	2.5
dirty buffers inspected	45,999	36.7	0.7
enqueue conversions	24,380	19.4	0.4
enqueue releases	288,303	229.7	4.3
enqueue requests	290,599	231.6	4.3
enqueue timeouts	2,220	1.8	0.0
enqueue waits	0	0.0	0.0
exchange deadlocks	107	0.1	0.0
execute count	15,969,809	12,725.0	237.2
free buffer inspected	66,008	52.6	1.0
free buffer requested	1,212,712	966.3	18.0

hot buffers moved to head of LRU	386,534	308.0	5.7
immediate (CR) block cleanout app	532,040	423.9	7.9
immediate (CURRENT) block cleanou	27,197	21.7	0.4
leaf node splits	7,232	5.8	0.1
logons cumulative	4,201	3.4	0.1
messages received	142,196	113.3	2.1
messages sent	142,198	113.3	2.1

Instance Activity Stats for DB: PRD Instance: PRD Snaps: 1 -2

Statistic	Total	per Second	per Trans
-----	-----	-----	-----
no buffer to keep pinned count	325,374,689	259,262.7	4,833.2
no work - consistent read gets	49,695,401	39,597.9	738.2
opened cursors cumulative	880,852	701.9	13.1
parse count (hard)	12,077	9.6	0.2
parse count (total)	3,724,799	2,968.0	55.3
parse time cpu	38,101	30.4	0.6
parse time elapsed	43,120	34.4	0.6
physical reads	243,512	194.0	3.6
physical reads direct	16,254	13.0	0.2
physical writes	168,054	133.9	2.5
physical writes direct	16,921	13.5	0.3
physical writes non checkpoint	154,000	122.7	2.3
pinned buffers inspected	68	0.1	0.0
prefetched blocks	532	0.4	0.0
prefetched blocks aged out before	7	0.0	0.0
process last non-idle time	#####	#####	#####
recursive calls	25,653,704	20,441.2	381.1
recursive cpu usage	863,233	687.8	12.8
redo blocks written	1,706,143	1,359.5	25.3
redo entries	2,723,829	2,170.4	40.5
redo ordering marks	0	0.0	0.0
redo size	835,182,028	665,483.7	12,406.0
redo synch time	108,456	86.4	1.6
redo synch writes	50,462	40.2	0.8

redo wastage	10,678,260	8,508.6	158.6
redo write time	15,383	12.3	0.2
redo writer latching time	260	0.2	0.0
redo writes	42,803	34.1	0.6
rollback changes - undo records a	64	0.1	0.0
rollbacks only - consistent read	317,311	252.8	4.7
rows fetched via callback	132,604,071	105,660.6	1,969.7
session connect time	#####	#####	#####
session cursor cache hits	3,345,492	2,665.7	49.7
session logical reads	502,695,834	400,554.5	7,467.2
session pga memory	5,904,629,900	4,704,884.4	87,708.6
session pga memory max	6,526,554,040	5,200,441.5	96,946.8
session uga memory max	3,810,406,788	3,036,180.7	56,600.6
sorts (disk)	68	0.1	0.0
sorts (memory)	1,688,908	1,345.7	25.1
sorts (rows)	63,070,601	50,255.5	936.9
summed dirty queue length	15,536	12.4	0.2
switch current to new buffer			
table fetch by rowid	286,582,891	228,352.9	4,257.0
table fetch continued row	1,835	1.5	0.0
table scan blocks gotten	6,092,998	4,855.0	90.5
table scan rows gotten	154,736,111	123,295.7	2,298.5
table scans (long tables)	0	0.0	0.0
table scans (short tables)	3,364,201	2,680.6	50.0
total file opens	4,047	3.2	0.1
transaction rollbacks	16	0.0	0.0
user calls	8,916,155	7,104.5	132.4
user commits	46,826	37.3	0.7
user rollbacks	20,495	16.3	0.3
write clones created in backgroun	0	0.0	0.0
write clones created in foregroun	13	0.0	0.0

Tablespace IO Stats for DB: PRD Instance: PRD Snaps: 1 -2

->ordered by IOs (Reads + Writes) desc

Tablespace

```

-----
              Av      Av      Av              Av      Buffer Av Buf
              Reads Reads/s Rd(ms) Blks/Rd      Writes Writes/s      Waits Wt(ms)
-----
USER_IDX
    111,990      89      7.4      1.0      71,063      57      2,182      40.8
USER_DATA
    112,500      90      8.9      1.0      30,373      24      535      10.9
RBSRD
    151          0     16.0      1.0      16,305      13      2,467      2.2
RBSND
    177          0     11.4      1.0      15,055      12      2,823      1.8
RBS
    162          0     18.8      1.0      13,484      11      2,766      2.3
SYSTEM
    233          0      9.1      1.3      3,790       3      175      12.1
TEMP
    832          1      0.0     16.3      2,446       2          0      0.0
-----

```

File IO Stats for DB: PRD Instance: PRD Snaps: 1 -2
->ordered by Tablespace, File

```

Tablespace      Filename
-----
              Av      Av      Av              Av      Buffer Av Buf
              Reads Reads/s Rd(ms) Blks/Rd      Writes Writes/s      Waits Wt(ms)
-----
RBS              /d02/user_rbs01.dbf
    162          0     18.8      1.0      13,484      11      2,766      2.3
RBSND            /d02/user_rbs02.dbf
    177          0     11.4      1.0      15,055      12      2,823      1.8
RBSRD            /d02/user_rbs03.dbf
    151          0     16.0      1.0      16,305      13      2,467      2.2

```

SYSTEM			/d02/sys01.dbf						
	233	0	9.1	1.3	3,790	3	175	12.1	
TEMP			/d02/user_temp01.dbf						
	832	1	0.0	16.3	2,446	2	0		
USER_DATA			/d02/user_data01.dbf						
	8,138	6	4.3	1.0	9,819	8	93	13.5	
			/d02/user_data02.dbf						
	10,090	8	11.2	1.0	1,974	2	163	13.6	
			/d02/user_data03.dbf						
	22,551	18	9.6	1.0	561	0	0		
			/d02/user_data04.dbf						
	5,632	4	10.4	1.1	2,680	2	73	9.3	
			/d02/user_data05.dbf						
	7,863	6	3.7	1.0	10,498	8	109	3.9	
			/d02/user_data06.dbf						
	44,250	35	9.2	1.0	3,301	3	14	34.3	
			/d02/user_data07.dbf						
	7,817	6	10.8	1.0	324	0	2	10.0	
			/d02/user_data08.dbf						
	6,159	5	9.8	1.0	1,216	1	81	9.5	
USER_IDX			/d02/user_idx01.dbf						
	27,809	22	5.5	1.0	27,993	22	270	4.3	
			/d02/user_idx02.dbf						
	27,639	22	7.3	1.0	10,716	9	37	29.7	
			/d02/user_idx03.dbf						
	969	1	10.6	1.0	859	1	26	2.3	
			/d02/user_idx04.dbf						
	7,110	6	11.0	1.0	4,306	3	401	9.6	
			/d02/user_idx05.dbf						
	43,164	34	8.2	1.0	19,068	15	1,219	66.8	
			/d02/user_idx06.dbf						
	5,299	4	5.5	1.0	8,121	6	229	5.5	

Buffer Pool Statistics for DB: PRD Instance: PRD Snaps: 1 -2

-> Pools D: default pool, K: keep pool, R: recycle pool

	Buffer Gets	Consistent Gets	Physical Reads	Physical Writes	Free Buffer Waits	Write Complete Waits	Buffer Busy Waits
D	1,205,282	0	225,751	150,070	0	0	10,950

Buffer wait Statistics for DB: PRD Instance: PRD Snaps: 1 -2

-> ordered by wait time desc, waits desc

Class	Waits	Tot Wait Time (cs)	Avg Time (cs)
data block	2,926	9,743	3
undo block	7,309	1,516	0
undo header	834	178	0
segment header	9	0	0

Rollback Segment Stats for DB: PRD Instance: PRD Snaps: 1 -2

->A high value for "Pct Waits" suggests more rollback segments may be required

RBS No	Trans Table Gets	Pct Waits	Undo Bytes Written	Wraps	Shrinks	Extends
0	7.0	0.00	0	0	0	0
1	619.0	0.00	979,040	0	0	0
2	678.0	0.00	1,050,642	0	0	0
3	732.0	0.00	1,102,162	0	0	0
...						
297	8.0	0.00	5,656	0	0	0

Rollback Segment Storage for DB: PRD Instance: PRD Snaps: 1 -2
 ->Optimal Size should be larger than Avg Active

RBS No	Segment Size	Avg Active	Optimal Size	Maximum Size
0	13,082,624	8,192		13,082,624
1	21,291,008	617,676		21,291,008
2	21,291,008	212,992		21,291,008
3	21,291,008	617,676		21,291,008
....				
297	21,291,008	212,992		21,291,008

Latch Activity for DB: PRD Instance: PRD Snaps: 1 -2

->"Get Requests", "Pct Get Miss" and "Avg Slps/Miss" are statistics for willing-to-wait latch get requests
 ->"NoWait Requests", "Pct NoWait Miss" are for no-wait latch get requests
 ->"Pct Misses" for both should be very close to 0.0

Latch Name	Get Requests	Pct Get Miss	Avg Slps /Miss	NoWait Requests	Pct NoWait Miss
Token Manager	4,051	0.0		0	
active checkpoint queue latch	99,841	0.0		0	
cache buffer handles	34,806	0.0	0.0	0	
cache buffers chains	683,179,213	2.8	0.0	1,202,253	1.7
cache buffers lru chain	12,583,508	0.1	0.0	1,104,002	0.1
channel handle pool latch	1,198	0.0		1,193	0.0
channel operations parent lat	2,396	0.0	2.0	1,193	0.0
checkpoint queue latch	10,169,082	0.1	0.0	0	
dictionary lookup	125	0.0		0	
dml lock allocation	400,169	0.1	0.0	0	
enqueue hash chains	604,976	0.1	0.0	0	
enqueues	487,642	0.1	0.0	0	

event group latch	1,198	0.0		0	
job_queue_processes parameter	24	0.0		0	
ktm global data	5	0.0		0	
latch wait list	22,805	14.0	0.0	20,459	0.6
library cache	96,789,917	3.0	0.0	0	
library cache load lock	376	0.0		0	
list of block allocation	175,119	0.0	0.0	0	
loader state object freelist	482	0.0		0	
messages	553,582	0.2	0.0	0	
multiblock read objects	4,806	0.0		0	
ncodef allocation latch	24	0.0		0	
process allocation	1,198	2.5	1.0	1,198	0.0
process group creation	2,391	0.0		0	
redo allocation	2,816,298	4.2	0.0	0	
redo writing	344,566	0.4	0.0	0	
row cache objects	2,407,917	0.3	0.0	0	
sequence cache	616,884	0.2	0.0	0	
session allocation	911,774	0.3	0.0	0	
session idle bit	18,008,054	0.0	0.0	0	
session switching	12,062	0.0	0.0	0	
shared pool	6,413,300	5.0	0.0	0	
sort extent pool	407	0.0		0	
transaction allocation	239,890	0.2	0.0	0	
transaction branch allocation	24	0.0		0	
undo global data	26,174,193	1.5	0.0	0	
user lock	16,880	0.1	0.0	0	

Latch Sleep breakdown for DB: PRD Instance: PRD Snaps: 1 -2

-> ordered by misses desc

Latch Name	Get		Spin &	
	Requests	Misses	Sleeps	Sleeps 1->4
-----	-----	-----	-----	-----
cache buffers chains	683,179,213	18,914,409	14,443	18901441/121 01/577/290/0

library cache	96,789,917	2,915,511	28,968	2895400/1400 5/4353/1753/ 0
undo global data	26,174,193	385,640	334	385327/293/1 9/1/0
shared pool	6,413,300	317,706	395	317325/368/1 2/1/0
redo allocation	2,816,298	117,203	22	117181/22/0/ 0/0
row cache objects	2,407,917	7,541	120	7422/118/1/0 /0
cache buffers lru chain	12,583,508	7,307	3	7304/3/0/0/0
session idle bit	18,008,054	5,609	58	5555/50/4/0/ 0
checkpoint queue latch	10,169,082	5,505	1	5504/1/0/0/0
session allocation	911,774	3,101	3	3098/3/0/0/0
redo writing	344,566	1,355	1	1354/1/0/0/0
sequence cache	616,884	1,201	1	1200/1/0/0/0
process allocation	1,198	30	30	0/30/0/0/0
channel operations parent	2,396	1	2	0/0/1/0/0

Latch Miss Sources for DB: PRD Instance: PRD Snaps: 1 -2

-> only latches with sleeps are shown

-> ordered by name, sleeps desc

Latch Name	Where	NoWait Misses	Waiter Sleeps	Waiter Sleeps
cache buffers chains	kcbgtcr: kslbegin	0	12,661	8,453
cache buffers chains	kcbrls: kslbegin	0	629	391
cache buffers chains	kcbzwb	0	525	95
cache buffers chains	kcbchg: kslbegin: bufs not	0	277	3,950
cache buffers chains	kcbchg: kslbegin: call CR	0	194	1,060
cache buffers chains	kcbgcur: kslbegin	0	115	317
cache buffers chains	kcbget: pin buffer	0	10	111

cache buffers chains	kcbget: exchange rls	0	6	1
cache buffers chains	kcbcge	0	4	24
cache buffers chains	kcbzgb: scan from tail. no	0	2	0
cache buffers lru chain	kcbzgb: multiple sets nowa	0	3	0
channel operations paren	ksrchdelete()	0	2	0
checkpoint queue latch	kcbk0rrd: update recovery	0	1	0
library cache	kglhdgn: child:	0	12,051	958
library cache	kglpnl: child: before pro	0	5,364	10,719
library cache	kglidp: parent	0	5,008	0
library cache	kglic	0	2,939	24
library cache	kglpin	0	899	922
library cache	kglget: child: KGLDSBRD	0	777	123
library cache	kglpnl: child: alloc spac	0	649	3,903
library cache	kglpnp: child	0	321	9,280
library cache	kgllkdl: child: cleanup	0	234	93
library cache	kglget: child: KGLDSBYD	0	216	1,405
library cache	kglupc: child	0	153	202
library cache	kglhdgc: child:	0	141	156
library cache	kgllkdl: child: free pin	0	59	408
library cache	kgldti: 2child	0	39	96
library cache	kglpnl: parent held, no p	0	30	0
library cache	kglpnc: child	0	18	447
library cache	kgldrp: parent	0	11	2
library cache	kglpndl: parent: purge	0	11	7
library cache	kgldte: child 0	0	6	194
process allocation	ksuapc	0	30	30
redo allocation	kcrfwr: redo allocation	0	22	21
redo writing	kcrfsr	0	1	1
row cache objects	kqrpre: find obj	0	61	80
row cache objects	kqrpsc: incr stat	0	59	0
sequence cache	kdnssd	0	1	0
session allocation	ksuprc	0	2	5
session allocation	ksudlc	0	1	1
session idle bit	ksupuc: clear busy	0	50	8
session idle bit	ksupuc: set busy	0	8	50
shared pool	kghalo	0	149	43

shared pool	kghupr1	0	119	333
shared pool	kghfre	0	97	18
shared pool	kghfnd: req scan	0	26	0
shared pool	kghfnd: min scan	0	4	0
undo global data	ktudba: KSLBEGIN	0	319	319
undo global data	ktubnd	0	15	7

Dictionary Cache Stats for DB: PRD Instance: PRD Snaps: 1 -2

->"Pct Misses" should be very low (< 2% in most cases)

->"Cache Usage" is the number of cache entries being used

->"Pct SGA" is the ratio of usage to allocated size for that cache

Cache	Get Requests	Pct Miss	Scan Requests	Pct Miss	Mod Req	Final Usage	Pct SGA
dc_constraints	756	33.3	0		756	1	6
dc_database_links	0		0		0	0	0
dc_files	0		0		0	0	0
dc_free_extents	8,226	14.2	2,866	0.0	3,266	590	97
dc_global_oids	0		0		0	0	0
dc_histogram_data	0		0		0	0	0
dc_histogram_data_valu	0		0		0	0	0
dc_histogram_defs	567	100.0	0		0	266	90
dc_object_ids	183,147	0.1	0		105	1,752	100
dc_objects	125,806	0.1	0		305	2,288	100
dc_outlines	0		0		0	0	0
dc_profiles	4,181	0.0	0		0	1	13
dc_rollback_segments	3,588	0.0	0		0	300	97
dc_segments	218,767	0.0	0		1,283	2,086	100
dc_sequence_grants	22,074	0.0	0		0	271	100
dc_sequences	11,034	0.0	0		9,336	42	88
dc_synonyms	43,977	0.1	0		84	478	98
dc_tablespace_quotas	4,383	0.1	0		1,146	19	83
dc_tablespaces	3,847	0.0	0		0	6	86
dc_used_extents	1,190	91.9	0		1,190	5,017	100
dc_user_grants	53,259	0.0	0		0	61	57

dc_usernames	61,910	0.0	0	0	24	89
dc_users	71,102	0.0	0	0	70	89
ifs_acl_cache_entries	0		0	0	0	0

Library Cache Activity for DB: PRD Instance: PRD Snaps: 1 -2

->"Pct Misses" should be very low

Namespace	Get Requests	Pct Miss	Pin Requests	Pct Miss	Reloads	Invali- dations
BODY	57,683	0.0	57,673	0.0	0	0
CLUSTER	0		0		0	0
INDEX	84	50.0	84	50.0	0	0
OBJECT	0		0		0	0
PIPE	626,110	-0.0	626,106	-0.0	0	0
SQL AREA	29,805	1.3	17,421,119	-0.0	11,343	231
TABLE/PROCEDURE	168,117	0.1	11,275,144	-0.1	42	0
TRIGGER	14,983	0.0	14,983	0.0	0	0

SGA Memory Summary for DB: PRD Instance: PRD Snaps: 1 -2

SGA regions	Size in Bytes
Database Buffers	1,073,741,824
Fixed Size	73,888
Redo Buffers	2,113,536
Variable Size	1,896,189,952
sum	2,972,119,200

SGA breakdown difference for DB: PRD		Instance: PRD	Snaps: 1 -2	
Pool	Name	Begin value	End value	Difference
java pool	free memory	20,000,768	20,000,768	0
shared pool	Checkpoint queue	1,729,728	1,729,728	0
shared pool	KGFF heap	14,200	14,200	0
shared pool	KGK heap	9,176	9,176	0
shared pool	KQLS heap	10,593,700	10,614,840	21,140
shared pool	PL/SQL DIANA	6,827,692	6,827,692	0
shared pool	PL/SQL MPCODE	12,106,928	12,106,928	0
shared pool	PLS non-lib hp	2,892	2,892	0
shared pool	SEQ S.O.	2,728,000	2,728,000	0
shared pool	State objects	30,066,424	30,066,424	0
shared pool	branches	8,184,240	8,184,240	0
shared pool	db_block_buffers	17,825,792	17,825,792	0
shared pool	db_block_hash_buckets	8,912,952	8,912,952	0
shared pool	db_handles	10,500,000	10,500,000	0
shared pool	dictionary cache	3,705,820	3,981,228	275,408
shared pool	distributed_transactions	3,069,152	3,069,152	0
shared pool	enqueue_resources	2,949,120	2,949,120	0
shared pool	event statistics per ses	107,632,000	107,632,000	0
shared pool	fixed allocation callbac	5,732	5,732	0
shared pool	free memory	525,352,380	502,363,120	-22,989,260
shared pool	ktlbnk state objects	14,594,800	14,594,800	0
shared pool	library cache	694,582,988	697,977,328	3,394,340
shared pool	messages	3,696,000	3,696,000	0
shared pool	miscellaneous	98,497,740	98,343,228	-154,512
shared pool	processes	16,716,000	16,716,000	0
shared pool	sessions	66,836,000	66,836,000	0
shared pool	simulator trace entries	960,000	960,000	0
shared pool	sql area	182,260,376	201,115,500	18,855,124
shared pool	state objects	9,012,280	9,608,680	596,400
shared pool	table columns	23,064	23,624	560
shared pool	table definiti	15,284	16,084	800
shared pool	transaction_branches	6,274,400	6,274,400	0
shared pool	transactions	30,417,200	30,417,200	0

shared pool trigger defini	52,188	52,188	0
shared pool trigger inform	240	240	0
shared pool view columns d	916	916	0
db_block_buffers	1,073,741,824	1,073,741,824	0
fixed_sga	73,888	73,888	0
log_buffer	2,097,152	2,097,152	0

init.ora Parameters for DB: PRD Instance: PRD Snaps: 1 -2

Parameter Name	Begin value	End value (if different)
-----	-----	-----
_db_block_hash_buckets	1048576	
_db_block_hash_latches	4096	
_db_writer_chunk_writes	50	
_db_writer_max_writes	256	
_spin_count	16000	
background_dump_dest	/d01/app/oracle/admin/PRD/bdump	
compatible	8.1.7	
control_files	/d02/ctl01.ctl, /d02/ctl02.ctl	
core_dump_dest	/d01/app/oracle/admin/PRD/cdump	
cursor_sharing	FORCE	
cursor_space_for_time	TRUE	
db_block_buffers	131072	
db_block_lru_latches	48	
db_block_max_dirty_target	65536	
db_block_size	8192	
db_file_multiblock_read_count	128	
db_files	500	
db_name	PRD	
dml_locks	6000	
enqueue_resources	40960	
hash_multiblock_io_count	4	
log_buffer	2097152	
log_checkpoint_interval	1410065407	
log_checkpoints_to_alert	TRUE	
max_dump_file_size	102400	

max_enabled_roles	100
nls_date_format	DD-MON-RR
nls_numeric_characters	.,
nls_sort	binary
open_cursors	4000
optimizer_mode	RULE
processes	21000
row_locking	always
session_cached_cursors	200
sessions	31000
shared_pool_reserved_size	67108864
shared_pool_size	1610612736
sort_area_retained_size	0
sort_area_size	1024000
sql_trace	FALSE
timed_statistics	TRUE
user_dump_dest	/d01/app/oracle/admin/PRD/udump
utl_file_dir	/usr/tmp, /tmp

Appendix C: report.txt to Show Buffer Busy Waits

```

SVRMGR>
SVRMGR> set charwidth 12
Charwidth                12
SVRMGR> set numwidth 10
Numwidth                 10
SVRMGR> Rem Select Library cache statistics.  The pin hit rate should be high.
SVRMGR> select namespace library,
2>      gets,
3>      round(decode(gethits,0,1,gethits)/decode(gets,0,1,gets),3)
4>      gethitratio,
5>      pins,
6>      round(decode(pinhits,0,1,pinhits)/decode(pins,0,1,pins),3)

```

```

7>         pinhitratio,
8>         reloads, invalidations
9> from stats$lib;

```

LIBRARY	GETS	GETHITRATI	PINS	PINHITRATI	RELOADS	INVALIDATI
BODY	4856	1	4856	1	0	0
CLUSTER	18	.944	30	.933	0	0
INDEX	0	1	0	1	0	0
OBJECT	0	1	0	1	0	0
PIPE	0	1	0	1	0	0
SQL AREA	188546	.998	581484	.999	0	0
TABLE/PROCED	656	.698	7084	.933	0	0
TRIGGER	0	1	0	1	0	0

```
8 rows selected.
```

```
SVRMGR>
```

```
SVRMGR> set charwidth 27;
```

```
Charwidth                27
```

```
SVRMGR> set numwidth 12;
```

```
Numwidth                  12
```

```
SVRMGR> Rem The total is the total value of the statistic between the time
```

```
SVRMGR> Rem bstat was run and the time estat was run. Note that the estat
```

```
SVRMGR> Rem script logs on as "internal" so the per_logon statistics will
```

```
SVRMGR> Rem always be based on at least one logon.
```

```
SVRMGR> select n1.name "Statistic",
```

```
2>         n1.change "Total",
```

```
3>         round(n1.change/trans.change,2) "Per Transaction",
```

```
4>         round(n1.change/(to_number(to_char(end_time, 'J'))*60*60*24 -
```

```
5>                 to_number(to_char(start_time, 'J'))*60*60*24 +
```

```
6>                 to_number(to_char(end_time, 'SSSSS'))
```

```
7>                 -
to_number(to_char(start_time,
'SSSSS'))))
```

```
8>                 , 2) "Per Second"
```

```
9> from stats$stats n1, stats$stats trans, stats$dates
```

```
10> where trans.name='user commits'
```

```
11> and n1.change != 0
```

```

12> order by nl.name;
Statistic                               Total           Per Transact Per Second
-----
CPU used by this session                 114518598        420.14      33426.33
CPU used when call started               6506491          23.87       1899.15
CR blocks created                        2312253          8.48        674.91
DBWR buffers scanned                    329020583       1207.08     96036.36
DBWR checkpoint buffers wri             141183           .52         41.21
DBWR checkpoint write reque             8832             .03         2.58
DBWR checkpoints                         2                0           0
DBWR free buffers found                  307374700       1127.67     89718.24
DBWR lru scans                           249436           .92         72.81
DBWR make free requests                  240604           .88         70.23
DBWR revisited being-writte             171              0           .05
DBWR skip hot writes                     128              0           .04
DBWR summed scan depth                   329119778       1207.45     96065.32
DBWR transaction table writ              4657            .02         1.36
DBWR undo block writes                   3417639          12.54       997.56
SQL*Net roundtrips to/from              63480027         232.89     18528.9
background checkpoints comp               2                0           0
background checkpoints star               2                0           0
background timeouts                       16137            .06         4.71
buffer is not pinned count                105474299        386.96     30786.43
buffer is pinned count                    70044849         256.97     20445.08
bytes received via SQL*Net               13680342497     50189.28   3993094.72
bytes sent via SQL*Net to c              21044030582     77204.55   6142449.09
calls to get snapshot scn:               65429138         240.04     19097.82
calls to kcmgas                           1444928           5.3        421.75
calls to kcmgcs                           4181133          15.34     1220.41
calls to kcmgrs                           45907874         168.42     13399.85
change write time                          726125           2.66       211.95
cleanouts and rollbacks - c              1666450           6.11       486.41
cleanouts only - consistent               408784            1.5        119.32
cluster key scan block gets              11934745          43.79     3483.58
cluster key scans                         9329308           34.23     2723.09
commit cleanout failures: b              717824            2.63       209.52

```

commit cleanout failures: b	15090	.06	4.4
commit cleanout failures: c	5672	.02	1.66
commit cleanout failures: c	25328	.09	7.39
commit cleanouts	12650303	46.41	3692.44
commit cleanouts successful	11886389	43.61	3469.47
consistent changes	22114773	81.13	6454.98
consistent gets	340090928	1247.7	99267.64
cursor authentications	692	0	.2
data blocks consistent read	22088980	81.04	6447.45
db block changes	158143696	580.18	46159.86
db block gets	161361949	591.99	47099.23
deferred (CURRENT) block cl	2578600	9.46	752.66
dirty buffers inspected	102234	.38	29.84
enqueue conversions	212819	.78	62.12
enqueue releases	2634484	9.67	768.97
enqueue requests	2635447	9.67	769.25
enqueue timeouts	683	0	.2
enqueue waits	10316	.04	3.01
exchange deadlocks	5792	.02	1.69
execute count	63205800	231.88	18448.86
free buffer inspected	180154	.66	52.58
free buffer requested	25295016	92.8	7383.25
immediate (CR) block cleano	2075235	7.61	605.73
immediate (CURRENT) block c	3810730	13.98	1112.3
logons cumulative	5538	.02	1.62
logons current	-1	0	0
messages received	386612	1.42	112.85
messages sent	386612	1.42	112.85
no work - consistent read g	117449311	430.89	34281.76
opened cursors cumulative	210340	.77	61.4
opened cursors current	95236	.35	27.8
parse count (hard)	376	0	.11
parse count (total)	209813	.77	61.24
parse time cpu	6585	.02	1.92
parse time elapsed	446559	1.64	130.34
physical reads	17019443	62.44	4967.73

physical writes	14470023	53.09	4223.59
process last non-idle time	2247325781	8244.8	655961.99
recursive calls	1456115	5.34	425.02
recursive cpu usage	80607	.3	23.53
redo blocks written	112357950	412.21	32795.67
redo buffer allocation retr	237	0	.07
redo entries	82243100	301.73	24005.58
redo log space requests	243	0	.07
redo log space wait time	8006	.03	2.34
redo size	55640388840	204128.73	16240627.22
redo small copies	11303238	41.47	3299.25
redo synch time	741222	2.72	216.35
redo synch writes	266295	.98	77.73
redo wastage	81457724	298.85	23776.34
redo write time	150186	.55	43.84
redo writer latching time	42431	.16	12.38
redo writes	327340	1.2	95.55
rollback changes - undo rec	5741	.02	1.68
rollbacks only - consistent	612146	2.25	178.68
session connect time	2247325781	8244.8	655961.99
session cursor cache count	1218	0	.36
session cursor cache hits	16410	.06	4.79
session logical reads	479355907	1758.62	139917.08
session pga memory	3044461740	11169.26	888634.48
session pga memory max	3089992764	11336.3	901924.33
session uga memory	833068272	3056.29	243160.62
session uga memory max	1619159912	5940.24	472609.43
sorts (disk)	42	0	.01
sorts (memory)	3658616	13.42	1067.9
sorts (rows)	6607233	24.24	1928.56
summed dirty queue length	90331	.33	26.37
table fetch by rowid	85078281	312.13	24833.12
table fetch continued row	39785	.15	11.61
table scan blocks gotten	42616	.16	12.44
table scan rows gotten	315341	1.16	92.04
table scans (long tables)	1	0	0


```

SVRMGR> Rem 2. 1/4 of db_block_buffers
SVRMGR> Rem which ever is smaller and also there is a platform specific limit
SVRMGR> Rem on the write batch size (normally 1024 or 2048 buffers). If the average
SVRMGR> Rem length of the dirty buffer write queue is larger than the value
SVRMGR> Rem calculated before, increase db_file_simultaneous_writes or db_files.
SVRMGR> Rem Also check for disks that are doing many more IOs than other disks.
SVRMGR> select round(queue.change/greatest(1,writes.change),2) "Average Write
      Queue Length"
      2>   from stats$stats queue, stats$stats writes
      3>   where queue.name = 'summed dirty queue length'
      4>   and writes.name = 'write requests';
Average Write Queue Length
-----
                          .37
1 row selected.
SVRMGR>
SVRMGR>
SVRMGR> set charwidth 32;
Charwidth                  32
SVRMGR> set numwidth 13;
Numwidth                   13
SVRMGR> Rem System wide wait events for non-background processes (PMON,
SVRMGR> Rem SMON, etc). Times are in hundreths of seconds. Each one of
SVRMGR> Rem these is a context switch which costs CPU time. By looking at
SVRMGR> Rem the Total Time you can often determine what is the bottleneck
SVRMGR> Rem that processes are waiting for. This shows the total time spent
SVRMGR> Rem waiting for a specific event and the average time per wait on
SVRMGR> Rem that event.
SVRMGR> select                                nl.event "Event Name",
      2>                                       nl.event_count "Count",
      3>                                       nl.time_waited "Total Time",
      4>                                       round(nl.time_waited/nl.event_count, 2) "Avg
      Time"
      5>   from stats$event nl
      6>   where nl.event_count > 0
      7>   order by nl.time_waited desc;

```

Event Name	Count	Total Time	Avg Time
SQL*Net message from client	63480211	272167482	4.29
enqueue	671895	62187532	92.56
db file sequential read	16993220	30395198	1.79
buffer busy waits	1909257	2699731	1.41
latch free	740908	842286	1.14
log file sync	260185	738806	2.84
rdbms ipc message	5544	561517	101.28
library cache pin	8945	437854	48.95
free buffer waits	3370	329626	97.81
SQL*Net message to client	63480170	66985	0
write complete waits	3215	27219	8.47
buffer deadlock	5792	12825	2.21
log file switch completion	243	8006	32.95
SQL*Net more data to client	34819	7499	.22
file open	59493	5776	.1
library cache load lock	67	5066	75.61
undo segment extension	1236	2848	2.3
db file scattered read	1212	2109	1.74
index block split	326	771	2.37
refresh controlfile command	831	372	.45
direct path write	264	128	.48
control file sequential read	8459	85	.01
switch logfile command	1	23	23
SQL*Net break/reset to client	263	12	.05
rdbms ipc reply	9	5	.56

25 rows selected.

SVRMGR> This was my 1st cut at the reports that show bufferbusy waits

SVRMGR>

SVRMGR> Rem Specific buffer wait statistics. Assumption: catparr.sql was run and at least one snp process is running.

```
SVRMGR> select map.name segment_name, map.kind segment_typ,
          sum(b.seconds_in_wait) sum_wait_time
2>      from stats$bufferwaits b, extent_to_object map
```

```

3>   where b.pltext='file#' and b.pl=map.file# and b.p2 between map.lowb and
      map.highb
4>   group by map.name, map.kind
5>   order by sum_wait_time desc;

```

SEGMENT_NAME	SEGMENT_TYP	SUM_WAIT_TIME
-----	-----	-----
NRIV	TABLE	8036
CBKK92	CLUSTER	1612
BKK9B~0	INDEX	1258
BKKIT~1	INDEX	949
BKK9B	TABLE	913
BKK42~REF	INDEX	561
CCDCLS	CLUSTER	396
VBMOD	TABLE	372
VBDATA	TABLE	335
BKKIT~2	INDEX	302
BKKIT	TABLE	177
BKK9A	TABLE	175
BKKMDISPOBJECTS~0	INDEX	170
BKK96~0	INDEX	144
BKKM2	TABLE	139
BKKIT~3	INDEX	109
BKK45	TABLE	106
BKK96	TABLE	105
BKKMDISPOBJECTS~2	INDEX	104
BKK9A~0	INDEX	102
BKKM2~1	INDEX	93
BKKMDISPOBJECTS~1	INDEX	81
BKK45~0	INDEX	79
BKKE1	TABLE	78
BUT000	TABLE	68
BKKITGL~0	INDEX	65
BKKM2~0	INDEX	64
BKK42	TABLE	63
....		
BKK99~0	INDEX	0

762 rows selected.

SVRMGR>

SVRMGR>

SVRMGR> Rem System wide wait events for background processes (PMON, SMON, etc)

```
SVRMGR> select          nl.event "Event Name",
2>                    nl.event_count "Count",
3>                    nl.time_waited "Total Time",
4>                    round(nl.time_waited/nl.event_count, 2) "Avg
      Time"
5>  from stats$bck_event nl
6>  where nl.event_count > 0
7>  order by nl.time_waited desc;
```

Event Name	Count	Total Time	Avg Time
-----	-----	-----	-----
rdbms ipc message	93880	1999048	21.29
db file parallel write	244776	1462767	5.98
pmon timer	1194	342348	286.72
smon timer	11	330034	30003.09
log file parallel write	327456	150037	.46
log file sync	9999	54727	5.47
latch free	40823	45520	1.12
control file parallel write	1190	6815	5.73
db file sequential read	1553	2588	1.67
direct path read	1744	1101	.63
db file scattered read	229	726	3.17
file open	930	62	.07
control file sequential read	2143	16	.01
direct path write	456	8	.02
buffer busy waits	7	7	1
enqueue	2	4	2
log file single write	4	3	.75
log file sequential read	2	2	1
file identify	13	1	.08

19 rows selected.

SVRMGR>

SVRMGR>

```

SVRMGR> set charwidth 18;
Charwidth                18
SVRMGR> set numwidth 11;
Numwidth                 11
SVRMGR> Rem Latch statistics. Latch contention will show up as a large value for
SVRMGR> Rem the 'latch free' event in the wait events above.
SVRMGR> Rem Sleeps should be low. The hit_ratio should be high.
SVRMGR> select name latch_name, gets, misses,
      2>    round((gets-misses)/decode(gets,0,1,gets),3)
      3>    hit_ratio,
      4>    sleeps,
      5>    round(sleeps/decode(misses,0,1,misses),3) "SLEEPS/MISS"
      6> from stats$latches
      7> where gets != 0
      8> order by name;

```

LATCH_NAME	GETS	MISSES	HIT_RATIO	SLEEPS	SLEEPS/MISS
Active checkpoint	25926	57	.998	0	0
Checkpoint queue l	80887600	117222	.999	2310	.02
Token Manager	61093	79	.999	0	0
cache buffer handl	1081292	1055	.999	8	.008
cache buffers chai	1272178813	8962815	.993	378349	.042
cache buffers lru	5670527	49955	.991	2050	.041
dml lock allocatio	1373854	2712	.998	38	.014
enqueue hash chain	13289560	102609	.992	1130	.011
enqueuees	4472796	47910	.989	824	.017
ktm global data	11	0	1	0	0
latch wait list	7032	301	.957	8	.027
library cache	2200318	39012	.982	8087	.207
library cache load	981	0	1	0	0
list of block allo	3965019	14798	.996	444	.03
loader state objec	264	0	1	0	0
messages	1418113	5306	.996	8	.002
modify parameter v	5602	0	1	0	0
multiblock read ob	3080	0	1	0	0
ncodef allocation	91	0	1	0	0

process allocation	7	0	1	0	0
redo allocation	82897417	18838373	.773	286106	.015
redo copy	97619	89910	.079	35502	.395
redo writing	1432618	7446	.995	3960	.532
row cache objects	318518	175	.999	1	.006
sequence cache	510	0	1	0	0
session allocation	560088	988	.998	21	.021
session idle bit	127098064	67454	.999	1835	.027
session switching	90	0	1	0	0
shared pool	85012	84	.999	0	0
sort extent pool	1107	0	1	0	0
transaction alloca	4338346	29780	.993	699	.023
transaction branch	91	0	1	0	0
undo global data	27525610	2157117	.922	60349	.028
user lock	26	0	1	0	0

34 rows selected.

SVRMGR>

SVRMGR> set numwidth 16

Numwidth 16

SVRMGR> Rem Statistics on no_wait gets of latches. A no_wait get does not

SVRMGR> Rem wait for the latch to become free, it immediately times out.

SVRMGR> select name latch_name,

```

2>   immed_gets nowait_gets,
3>   immed_miss nowait_misses,
4>   round((immed_gets/(immed_gets+immed_miss)), 3)
5>   nowait_hit_ratio
6> from stats$latches
7> where immed_gets + immed_miss != 0
8> order by name;
```

LATCH_NAME	NOWAIT_GETS	NOWAIT_MISSES	NOWAIT_HIT_RATIO
cache buffers chai	54092342	79629	.999
cache buffers lru	67668607	868524	.987
latch wait list	7457	366	.953
process allocation	7	0	1
redo copy	81312025	904409	.989

5 rows selected.

SVRMGR>

SVRMGR> Rem Buffer busy wait statistics. If the value for 'buffer busy wait' in
SVRMGR> Rem the wait event statistics is high, then this table will identify
SVRMGR> Rem which class of blocks is having high contention. If there are high
SVRMGR> Rem 'undo header' waits then add more rollback segments. If there are
SVRMGR> Rem high 'segment header' waits then adding freelists might help. Check
SVRMGR> Rem v\$session_wait to get the addresses of the actual blocks having
SVRMGR> Rem contention.

```
SVRMGR> select class,count,time,round(time/count,2) avg_wait_time from
          stats$waitstat
          2> where count != 0
          3> order by count desc;
```

CLASS	COUNT	TIME	AVG_WAIT_TIME
data block	1800548	2640498	1.47
undo header	63582	42977	.68
segment header	15010	6292	.42
undo block	9917	3032	.31
free list	9800	10664	1.09

5 rows selected.

SVRMGR>

SVRMGR> set numwidth 10;

Numwidth 10

SVRMGR>

SVRMGR> set charwidth 39

Charwidth 39

SVRMGR> Rem The init.ora parameters currently in effect:

```
SVRMGR> select name, value from v$parameter where isdefault = 'FALSE'
          2> order by name;
```

NAME	VALUE
_bump_highwater_mark_count	1000
_db_block_hash_buckets	400000
_log_io_size	128
background_dump_dest	/oracle/BC2/saptrace/background

control_files	/oracle/BC2/sapraw/cntrl1BC2.dbf, /orac
core_dump_dest	/oracle/BC2/saptrace/background
cursor_space_for_time	TRUE
db_block_buffers	200000
db_block_lru_latches	64
db_block_size	8192
db_file_multiblock_read_count	16
db_files	254
db_name	BCA
db_writer_processes	8
dml_locks	15000
enqueue_resources	8000
event	10061 trace name context forever, level
fast_full_scan_enabled	TRUE
hash_join_enabled	FALSE
job_queue_interval	1
job_queue_processes	2
log_archive_dest	/oracle/BC2/saparch/BC2arch
log_archive_start	FALSE
log_buffer	62914560
log_checkpoint_interval	1705032704
log_checkpoints_to_alert	TRUE
max_rollback_segments	700
open_cursors	800
optimizer_index_cost_adj	10
optimizer_mode	CHOOSE
parallel_max_servers	256
processes	2000
replication_dependency_tracking	FALSE
row_cache_cursors	100
row_locking	always
session_cached_cursors	50
sessions	2400
shared_pool_size	255013680
sort_area_retained_size	262144
sort_area_size	2097152

```

sort_read_fac                16
spin_count                   20000
temporary_table_locks       200
timed_statistics             TRUE
transaction_auditing         FALSE
transactions_per_rollback_segment 1
user_dump_dest               /oracle/BC2/saptrace/usertrace

```

47 rows selected.

SVRMGR>

SVRMGR>

SVRMGR> set charwidth 15;

```
Charwidth                15
```

SVRMGR> set numwidth 8;

```
Numwidth                8
```

SVRMGR> Rem get_miss and scan_miss should be very low compared to the requests.

SVRMGR> Rem cur_usage is the number of entries in the cache that are being used.

SVRMGR> select * from stats\$dc

2> where get_reqs != 0 or scan_reqs != 0 or mod_reqs != 0;

NAME	GET_REQS	GET_MISS	SCAN_REQ	SCAN_MIS	MOD_REQS	COUNT	CUR_USAG
dc_tablespaces	16759	22	0	0	0	33	25
dc_free_extents	19601	5579	15669	0	5932	836	826
dc_segments	6423	295	0	0	5442	1298	1282
dc_rollback_seg	16573	0	0	0	0	619	605
dc_used_extents	5453	5444	0	0	5442	5518	5512
dc_tablespace_q	121	5	0	0	121	9	5
dc_users	5300	0	0	0	0	27	12
dc_user_grants	284	0	0	0	0	15	10
dc_objects	679	181	0	0	0	573	564
dc_synonyms	30	19	0	0	0	34	29
dc_usernames	208	0	0	0	0	21	3
dc_object_ids	4591	185	0	0	0	503	501
dc_sequences	3	0	0	0	3	15	5
dc_tablespaces	76	0	0	0	76	25	1
dc_profiles	6	0	0	0	0	5	1
dc_histogram_de	140	67	0	0	0	74	68

16 rows selected.

SVRMGR>

SVRMGR>

SVRMGR> set charwidth 15;

Charwidth 15

SVRMGR> set numwidth 10;

Numwidth 10

SVRMGR>

SVRMGR> /*

2> select decode(COMMAND_TYPE,2,'insert',3,'select',6,'update',7,'delete')
Command,

3> EXECUTIONS, DISK_READS, BUFFER_GETS, ROWS_PROCESSED, sql_text as object

4> from stats\$sqlarea

5> order by EXECUTIONS desc;

6> */

SVRMGR>

SVRMGR> /*

2> select decode(COMMAND_TYPE,2,'insert',3,'select',6,'update',7,'delete')
Command,

3> EXECUTIONS, DISK_READS, BUFFER_GETS, ROWS_PROCESSED, sql_text as object

4> from stats\$sqlarea

5> order by DISK_READS desc;

6> */

SVRMGR> /*

2> select decode(COMMAND_TYPE,2,'insert',3,'select',6,'update',7,'delete')
Command,

3> EXECUTIONS, DISK_READS, BUFFER_GETS, ROWS_PROCESSED, sql_text as object

4> from stats\$sqlarea

5> order by ROWS_PROCESSED desc;

6> */

SVRMGR>

SVRMGR> set charwidth 25

Charwidth 25

SVRMGR> Rem The times that bstat and estat were run.

SVRMGR> select to_char(start_time, 'dd-mon-yy hh24:mi:ss') start_time,

2> to_char(end_time, 'dd-mon-yy hh24:mi:ss') end_time,

3> round(to_number(end_time-start_time)*24*60) minutes

```

4> from stats$dates;
START_TIME          END_TIME          MINUTES
-----
27-aug-99 02:27:34 27-aug-99 03:24:40          57
1 row selected.
SVRMGR>
SVRMGR> set charwidth 75
Charwidth          75
SVRMGR> Rem Versions
SVRMGR> select * from v$version;
BANNER
-----
Oracle8 Enterprise Edition Release 8.0.5.1.0 - Production
PL/SQL Release 8.0.5.1.0 - Production
CORE Version 4.0.5.0.0 - Production
TNS for Solaris: Version 8.0.5.0.0 - Production
NLSRTL Version 3.3.2.0.0 - Production
5 rows selected.
SVRMGR>
SVRMGR> spool off;

```

Appendix D: Buffer Busy Wait Analysis Scripts.

SQL Script to create table and views needed by analysis scripts.

```

==== crtabs.sql =====
create table stats$bufferwaits
(timestamp date,
sid number,
seq# number,
event varchar2(64),
pltext varchar2(64),
pl number,

```

```

plraw raw(4),
p2text varchar2(64),
p2 number,
p2raw raw(4),
p3text varchar2(64),
p3 number,
p3raw raw(4),
wait_time number,
seconds_in_wait number,
state varchar2(19)
)
/

```

create or replace view extent_to_object as

```

select file_id file#
       , block_id lowb
       , block_id+blocks-1 highb
       , segment_name name
       , partition_name
       , segment_type kind
from dba_extents
/

```

SQL Script to sample buffer busy waits.

=== **sampletst.sql** =====

```

                                     insert (tstamp,sid,seq#,into stats$bufferwaits
select sysdate tstamp,
       sid,
       seq#,
       event,
       pltext,
       pl,

```

```

        plraw,
        p2text,
        p2,
        p2raw,
        p3text,
        p3,
        p3raw,
        wait_time,
        seconds_in_wait,
        state
from v$session_wait
where pltext='file#' or event='enqueue'
;

/* Could alternatly use dbms_job package to submit job to sample a set intervals.
   This method is perfered.
*/

/*
execute dbms_job.isubmit(42,'insert into stats$bufferwaits select
        sysdate,sid,seq#,event,pltext,p1,p1raw,p2text,p2,p2raw,p3text,p3,p3
        raw,wait_time,seconds_in_wait,state from v$session_wait where
        pltext=''file#' or event=''enqueue'';insert into stats$locks select
        * from dmllocks;commit;',sysdate,'sysdate+1/86400');
*/

```

Report script to summarize top5 waits

```
=== bufbusyhist.sql =====
```

```

select 'SUMMARY of polled wait events by object' Name from dual;
select
        substr(map.name,0,15) segment_name,
        substr(map.kind,0,10) segment_typ,
        substr(b.event,0,17) event,
        substr(sum(b.seconds_in_wait),0,8) secswait
from stats$bufferwaits b, extent_to_object map

```

```

where b.pltext='file#' and
      b.p1=map.file# and
      b.p2 between map.lowb and map.highb
group by map.name, map.kind, b.event
order by sum(b.seconds_in_wait) desc
;

select 'TOP five buffer busy waits by object' Name from dual;
drop table foo;

```

```

create table foo as
select
  substr(map.name,0,15) segment_name,
  substr(map.kind,0,10) segment_typ,
  substr(b.event,0,17) event,
  substr(sum(b.seconds_in_wait),0,8) secswait
from stats$bufferwaits b, extent_to_object map
where b.pltext='file#' and
      b.p1=map.file# and
      b.p2 between map.lowb and map.highb
group by map.name, map.kind, b.event
order by sum(b.seconds_in_wait) desc
/

```

```

select * from foo where rownum < 6;

```

```

select 'HISTOGRAM of buffer busy waits of #1 busiest object' Name from dual;
select
  to_char(tstamp,'DD.MM.YY HH24:MI:SS SSSSS') timestamp,
  substr(map.name,0,15) segment_name,
  substr(map.kind,0,10) segment_typ,
  substr(b.event,0,17) event,
  substr(sum(b.seconds_in_wait),0,5) secswait
from stats$bufferwaits b, extent_to_object map
where b.pltext='file#' and
      b.p1=map.file# and

```

```

        b.p2 between map.lowb and map.highb
group by tstamp, map.name, map.kind, b.event
having map.name in (select segment_name from foo where rownum < 2)
/

```

=== Sample Output bufbusyhist.sql =====

NAME

TOP five buffer busy waits by object

SEGMENT_NAME	SEGMENT_TY	EVENT	SECSWAIT
WCLUSTER	CLUSTER	buffer busy waits	326
C_USER#	CLUSTER	buffer busy waits	18
CCLUSTER	CLUSTER	db file sequentia	6
T1	ROLLBACK	write complete wa	4
SCLUSTER	CLUSTER	db file sequentia	3

NAME

HISTOGRAM of buffer busy waits of #1 busiest object

TIMESTAMP	SEGMENT_NAME	SEGMENT_TY	EVENT	SECSW
07.02.00 14:06:38 50798	WCLUSTER	CLUSTER	buffer busy waits	1
07.02.00 14:06:40 50800	WCLUSTER	CLUSTER	buffer busy waits	3
07.02.00 14:06:43 50803	WCLUSTER	CLUSTER	buffer busy waits	0
07.02.00 14:06:53 50813	WCLUSTER	CLUSTER	buffer busy waits	0
07.02.00 14:06:55 50815	WCLUSTER	CLUSTER	buffer busy waits	22
07.02.00 14:06:59 50819	WCLUSTER	CLUSTER	buffer busy waits	1
07.02.00 14:07:02 50822	WCLUSTER	CLUSTER	buffer busy waits	2

07.02.00	14:07:19	50839	WCLUSTER	CLUSTER	buffer busy waits	1
07.02.00	14:07:22	50842	WCLUSTER	CLUSTER	buffer busy waits	0
07.02.00	14:07:30	50850	WCLUSTER	CLUSTER	buffer busy waits	2
07.02.00	14:07:32	50852	WCLUSTER	CLUSTER	buffer busy waits	1

TIMESTAMP		SEGMENT_NAME	SEGMENT_TY	EVENT	SECSW
07.02.00	14:07:37	50857	WCLUSTER	CLUSTER	buffer busy waits 6
07.02.00	14:07:41	50861	WCLUSTER	CLUSTER	buffer busy waits 1
07.02.00	14:07:43	50863	WCLUSTER	CLUSTER	buffer busy waits 3
07.02.00	14:07:45	50865	WCLUSTER	CLUSTER	buffer busy waits 0
07.02.00	14:07:47	50867	WCLUSTER	CLUSTER	buffer busy waits 2
07.02.00	14:07:52	50872	WCLUSTER	CLUSTER	buffer busy waits 0
07.02.00	14:07:54	50874	WCLUSTER	CLUSTER	buffer busy waits 3
07.02.00	14:08:00	50880	WCLUSTER	CLUSTER	buffer busy waits 1
07.02.00	14:08:02	50882	WCLUSTER	CLUSTER	buffer busy waits 3
07.02.00	14:08:04	50884	WCLUSTER	CLUSTER	buffer busy waits 6
07.02.00	14:08:07	50887	WCLUSTER	CLUSTER	buffer busy waits 13

TIMESTAMP		SEGMENT_NAME	SEGMENT_TY	EVENT	SECSW
07.02.00	14:08:09	50889	WCLUSTER	CLUSTER	buffer busy waits 8
07.02.00	14:08:13	50893	WCLUSTER	CLUSTER	buffer busy waits 1
07.02.00	14:08:17	50897	WCLUSTER	CLUSTER	buffer busy waits 1
07.02.00	14:08:20	50900	WCLUSTER	CLUSTER	buffer busy waits 3
07.02.00	14:08:22	50902	WCLUSTER	CLUSTER	buffer busy waits 0
07.02.00	14:08:24	50904	WCLUSTER	CLUSTER	buffer busy waits 6
07.02.00	14:08:26	50906	WCLUSTER	CLUSTER	buffer busy waits 1
07.02.00	14:08:28	50908	WCLUSTER	CLUSTER	buffer busy waits 0
07.02.00	14:08:35	50915	WCLUSTER	CLUSTER	buffer busy waits 0
07.02.00	14:08:37	50917	WCLUSTER	CLUSTER	buffer busy waits 0