



金蝶, 企业管理专家

# 性能调优-AWR 报表解读

EAS 技术支持部 马瑞琪

请注意：本文件只作为产品介绍之用，不属于您与金蝶签署的任何协议。本文件仅包括金蝶既定策略、产品及功能方面的信息，不能以本文件作为要求金蝶履行商务条款、产品策略以及开发义务的依据。本文件内容可能随时变更，恕不另行通知。

# 前言

## 概述

AWR 是 Oracle 10g 版本推出的新特性，全名叫 Automatic Workload Repository-自动负载信息库,AWR 是通过对比两次快照(snapshot)收集到的统计信息，来生成报表数据，生成的报表包括多个部分。




## 适用范围

现场实施同事、总部研发。

## 约定

### 符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 注意	表示有潜在风险，如果忽视这些文本，可能导致操作错误、数据丢失、设备性能降低或不可预知的结果。
 温馨提示	表示可以帮助您获取更多相关信息或提示您通过其它方式也可以获得相同结果。
 说明	强调或补充说明所描述的信息。

### 内容约定

本处对文中通用信息作如下约定

- \$ORACLE\_BASE 表示 ORACLE 产品的根目录，即安装过程中输入的本地路径。
- \$ORACLE\_HOME 表示 ORACLE 安装根目录，即安装过程中输入的本地路径。

## 目录

一、报表头 .....	2
二、负载简档 .....	2
三、实例的效率 .....	2
四、五个最重要的等待事件 .....	3
五、首要的 SQL 语句 .....	5
六、实例活动统计数据 .....	6
七、表空间和文件 I/O 统计数据 .....	7

## 一、报表头

### WORKLOAD REPOSITORY report for

DB Name	DB Id	Instance	Inst num	Release	RAC	Host
ORCL	1216967589	orcl	1	10.2.0.1.0	NO	Y-FTP

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	4	10-6月 -09 21:00:38	22	14.0
End Snap:	14	11-6月 -09 07:00:13	22	14.5
Elapsed:		599.59 (mins)		
DB Time:		7.09 (mins)		

### Report Summary

#### Cache Sizes

	Begin	End		
Buffer Cache:	184M	172M	Std Block Size:	8K
Shared Pool Size:	80M	92M	Log Buffer:	6,972K

## 二、负载简档

#### Load Profile

	Per Second	Per Transaction
Redo size:	1,067.21	5,945.06
Logical reads:	51.21	285.28
Block changes:	6.07	33.82
Physical reads:	0.09	0.50
Physical writes:	0.36	2.01
User calls:	1.75	9.77
Parses:	1.81	10.06
Hard parses:	0.13	0.75
Sorts:	1.12	6.23
Logons:	0.04	0.20
Executes:	6.17	34.35
Transactions:	0.18	

% Blocks changed per Read:	11.86	Recursive Call %:	95.47
Rollback per transaction %:	4.96	Rows per Sort:	36.95

如果重做数据块增加，块更改变得频繁，以及每次读操作%BLOCKS增加，都意味着DML语句活动的增加；

如果SQL语句不再共享池中分析，就会呈现硬分析，硬分析超过100次/秒就意味着绑定变量的使用效率不高，应当使用CURSOR\_SHARING初始化参数；或者说明共享池大小有问题；

如果SQL语句在共享池中运行就会出现软分析，软分析超过300次/秒就说明应用程序效率不高，语句被反复分析，而不是对应每个会话应只分析语句一次，以保证高效率。

## 三、实例的效率

#### Instance Efficiency Percentages (Target 100%)

Buffer Nowait %:	100.00	Redo NoWait %:	100.00
Buffer Hit %:	99.83	In-memory Sort %:	100.00
Library Hit %:	96.33	Soft Parse %:	92.57
Execute to Parse %:	70.73	Latch Hit %:	99.99
Parse CPU to Parse Elapsed %:	94.93	% Non-Parse CPU:	95.10

**BUFFER NOWAIT%低于99：**这是一个对特定缓冲区的请求命中率，在内存中该缓冲区应该立即可用。如果命中率下降，

在BUFFER WAIT部分将发现当前存在热数据块的争用现象；

**BUFFER HIT%低于95：**这是一个对特定缓冲区的请求命中率，并且缓冲区位于内存中，而无需物理磁盘的IO操作。

如果你在访问时经常使用非选择性索引，它将使命中率很高，这将导致很多DBA误认为系统性能很好。

高命中率不说明系统性能一定高，但低命中率一定说明低性能；

**LIBRARY HIT%低于95：**较低的库命中率通常意味着SQL过早的被推出缓冲池（可能是因为缓冲池太小了）。较低的命中率

还意味着没有使用绑定变量或者一些其他问题造成SQL没有被重用；

**OLTP中的IN MEMORY SORT%低于95：**在一个OLTP系统中，调整PGA\_AGGREGATE\_TARGET, SORT\_AREA\_SIZE可以有效解决该问题；

**SOFT\_PARSE%低于95：**软分析低于80说明SQL没有被重用；

**LATCH HIT%低于99：**通常是个大问题，找到特定的门锁可以解决问题。

## 四、五个最重要的等待事件

Top 5 Timed Events

Event	Waits	Time(s)	Avg Wait(ms)	% Total Call Time	Wait Class
control file parallel write	11,925	378	32	88.8	System I/O
CPU time		355		83.4	
db file parallel write	6,917	87	13	20.6	System I/O
log file parallel write	8,479	83	10	19.5	System I/O
log file sync	4,029	40	10	9.4	Commit

首要等待事件是整个报表中最能揭示问题的部分，如果TIMED\_STATISTICS是TRUE，那么事件将按照等待的时间排序，否则将按照等待的数量排序。

Wait Events

- s - second
- cs - centisecond - 100th of a second
- ms - millisecond - 1000th of a second
- us - microsecond - 1000000th of a second
- ordered by wait time desc, waits desc (file events last)

Event	Waits	%Time-outs	Total Wait Time (s)	Avg wait (ms)	Waits /txn
control file parallel write	11,925	0.00	378	32	1.86
db file parallel write	6,917	0.00	87	13	1.07
log file parallel write	8,479	0.00	83	10	1.31
log file sync	4,029	0.00	40	10	0.62
control file sequential read	14,894	0.00	15	1	2.28
os thread startup	552	0.00	10	19	0.09
db file sequential read	1,771	0.00	6	3	0.27
db file scattered read	265	0.00	1	4	0.04
SQL*Net break/reset to client	3,002	0.00	1	0	0.46
log file switch completion	2	0.00	0	158	0.00
Data file init write	16	0.00	0	16	0.00
SQL*Net message to client	56,511	0.00	0	0	8.75
rdbrms ipc reply	224	0.00	0	1	0.03
read by other session	4	0.00	0	16	0.00
latch: shared pool	20	0.00	0	2	0.00
library cache load lock	4	0.00	0	8	0.00
log file single write	2	0.00	0	9	0.00
latch free	3	0.00	0	6	0.00

direct path write	72	0.00	0	0	0.01
latch: library cache	20	0.00	0	1	0.00
LGWR wait for redo copy	145	0.00	0	0	0.02
SGA: allocation forcing component growth	1	0.00	0	11	0.00
db file single write	1	0.00	0	8	0.00
library cache pin	1	0.00	0	6	0.00
SQL*Net more data from client	99	0.00	0	0	0.02
row cache lock	7	0.00	0	0	0.00
buffer busy waits	19	0.00	0	0	0.00
log file sequential read	2	0.00	0	0	0.00
direct path write temp	59	0.00	0	0	0.01
undo segment extension	22	90.91	0	0	0.00
direct path read	4	0.00	0	0	0.00
latch: redo allocation	1	0.00	0	0	0.00
latch: cache buffers chains	1	0.00	0	0	0.00
cursor: mutex S	1	0.00	0	0	0.00
SQL*Net message from client	56,511	0.00	107,015	1894	8.75
Streams AQ: waiting for messages in the queue	7,185	100.00	35,887	4995	1.11
virtual circuit status	1,199	100.00	35,861	29909	0.19
Streams AQ: qmn slave idle wait	1,282	0.00	35,822	27943	0.20
Streams AQ: qmn coordinator idle wait	2,590	50.50	35,822	13831	0.40
wait for unread message on broadcast channel	35,794	100.00	35,795	1000	5.54
jobq slave wait	11,145	99.59	33,346	2992	1.73
Streams AQ: waiting for time management or cleanup tasks	21	100.00	13,265	631668	0.00
class slave wait	20	100.00	100	4999	0.00
SGA: MMAN sleep for component shrink	41	63.41	0	8	0.01

**DB FILE SCATTERED READ:** 该等待事件意味着等待与全表扫描或快速全索引扫描有关，该指数过大说明缺少索引或者限制使用索引，这种情况也可能是正常的，因为执行全表扫描可能比索引扫描效率更高。

**DB\_FILE\_MULTIBLOCK\_READ\_COUNT**能够加快全扫描（但也可能影响ORACLE完成更多扫描）。可以将表或者索引分区，以便至扫描其中一部分。缓慢的磁盘IO也会导致该等待；

**DB FILE SEQUENTIAL READ:** 该等待事件通常指单一的数据块读操作，例如索引的读取，该值过大说明表的连接顺序很糟糕，或者使用了非选择性索引。在一个高事务量，做过较好调整的系统该数字通常较大。以排序的方式加载数据有助于范围扫描，还可以减少块读取的数量，分区也有帮助，分区可以排除一些块。注意非选择性索引会导致许多快读取。

**BUFFER BUSY WAITS IDS:** 当一个缓冲区以一种非共享方式被使用或者正被读入缓存时会出现这种等待，

**BUFFER BUSY WAIT**不应该高于1%；

**buffer busy/Segment Header:** 如果等待针对一个段的头信息，就增加空闲列表或者空闲列表组的数量，或者扩大PCTUSED与PCTFREE之间的间隔。

**buffer busy/Undo Header:** 如果等待是针对一个撤销请求的头信息，则可以通过增加会滚段或者增加撤销区域来解决；

**Buffer Busy/Undo Block:** 如果是针对一个撤销的数据块，应该适当加快提交（不能过快，否则会引起日志同步问题）或者使用更大的回滚段或者撤销区。需要减少驱动一致性读操作的表上的数据密集度或者增加DB\_CACHE\_SIZE；

**Buffer Busy/Data Block:** 如果等待是发生在一个数据块上，就可以将数据移动到另外一个数据块上，以避免这个热数据块，或者使用更小的数据块（以减少每个数据块的行数，降低他的热度）；

**Buffer Busy/Index Block:**

**Latch Free:** 门锁是底层的队列机制（更加准确得说是互斥机制），用于保护系统全局去SGA的共享内存结构。如果门锁不可用，就会记录一次门锁丢失。大多数的门锁丢失问题都与使用绑定变量失败（库缓存门锁和共享池门锁）、生成重做日志问题、缓存的征用问题以及缓存的热数据块有关系。门锁也与BUG有关，当门锁丢失率高于0.5%就应当到oracle.com/support查找一下METLINK的BUG报告；

**ENQUEUE:** 入列是保护共享资源的一种锁机制。这种锁保护共享资源，例如一条记录中的数据，以防止两个人同时更新同一条数据，排队机制是FIFO（ORACLE的栓锁机制不是FIFO）；

**Log File Switch:** 确保归档的磁盘未满，并且速度足够快。由于IO的原因，DBWR可能速度很慢。可能需要

增加更多更大的日志文件，并且如果DBWR存在问题，你可能需要增加DBWR进程数；

**LOG BUFFER SPACE:**数据库发生改变时，改变的块被复制到日志缓冲区。如果日志缓冲区不能足够快的写入重做日志，就会导致LOG BUFFER SPACE问题。当一次提交大量事务时也会产生这个问题。

这种等待出现在写日志缓冲区的速度快于LGWR写重做日志的速度，或者是日志切换太慢了，但通常不是因为日志缓冲区太小。可以增加日志文件的尺寸，或者使用更快的磁盘写日志，但最终手段可以增加日志缓冲区的尺寸。

**LOG FILE SYNC:**为了减少日志文件同步等待，可以尝试一次批量提交更多的记录而不是逐条提交，可以使用更快的磁盘或者裸设备或者RAID10 (而不是RAID5, RAID5写性能太差)；

**Global Cache CR Request:**当使用多个实例时（RAC）当一个实例等待另一个实例缓存的数据块时就会发生Global Cache CR Request；

**Log File Parallel Write:**将重做日志放到较快的磁盘上，不要使用RAID5, 将日志文件和数据文件分开存放；

**Direct Path Read:** ORACLE通常使用DIRECT PATH READS直接将数据块读入PGA，它可用于排序、并行查询和提前读操作。这里的时间并不总是反映实际等待时间。这通常是文件I/O的问题。使用异步I/O可以减少消耗时间，

但是不会减少等待时间；

**Direct Path Write:**

**Async Disk I/O:**ORACLE等待异步写操作完成，或者等待写入的异步从属。问题可能是DBWR, LGWR, ARCH, CKPT有关的I/O问题，但通常是某个文件I/O问题。

**IDEL EVENTS:**闲置时间保存在STATS\$IDLE\_EVENT表中。

## 五、首要的 SQL 语句

### SQL ordered by Gets

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- Total Buffer Gets: 1,942,312
- Captured SQL account for 36.9% of Total

Buffer Gets	Executions	Gets per Exec	%Total	CPU Time (s)	Elapsed Time (s)	SQL ID	SQL Module	SQL Text
153,477	1	153,477.00	8.33	0.50	1.01	9f9v82h2v12h		select c.owner# in table_desc...
119,143	499	238.76	6.47	13.15	21.62	9qgch3vch3v3h		DECLARE job BRARY_INTEGER = ...
71,032	14,296	4.97	3.86	12.30	12.20	9b75vch3vch3h	OEM SystemPool	begin MGMT_JOB_ENGINE_get_sche...
46,325	796	58.20	2.51	3.05	3.06	9b8k7vch3vch3h	OEM SystemPool	INSERT INTO MGMT_METRICS_RAANC...
33,465	1,581	28.34	1.82	174.20	177.72	9b8k7vch3vch3h	OEM SystemPool	BEIGN END_NOTIFICATION_QUEUE_R...
28,697	8,926	3.22	1.56	1.29	1.50	9b8k7vch3vch3h	OEM SystemPool	UPDATE MGMT_CURRENT_METRICS SE...
26,915	14,296	1.89	1.40	1.40	1.40	9b8k7vch3vch3h	OEM SystemPool	SELECT JOB_ID, EXCURSION_ID...
25,529	599	44.29	1.44	2.63	2.64	9b8k7vch3vch3h	OEM SystemPool	BEIGN EM_PING_RECORD_BATCH_HEA...
25,816	361	71.51	1.40	2.30	2.95	9b8k7vch3vch3h		CALL MGMT_ADMIN_DATA_EVALUATE...
25,279	2,101	12.03	1.37	1.22	1.60	9b8k7vch3vch3h		insert into histgmlobj_int...
24,926	20	1,246.30	1.35	0.74	1.02	9b8k7vch3vch3h		select OBUOD, CLSOD, RUNT...
23,341	7,735	3.02	1.27	0.51	0.51	9b8k7vch3vch3h		select Pn rule 'Tracket_cnf...
22,980	2,916	8.77	1.26	0.71	0.72	9b8k7vch3vch3h		insert into sys_wnd_optmst_jn...
20,626	2,147	9.61	1.12	0.56	0.56	9b8k7vch3vch3h		select privilege_level from...
19,861	2,563	8.63	1.01	0.76	1.02	9b8k7vch3vch3h		select iobj, iobj, iobj...
19,426	6,470	2.85	1.00	0.97	0.97	9b8k7vch3vch3h		select job, m422nat_086...

如果磁盘读操作不是很高，但是缓冲区读操作很高而执行次数较低，说明查询可能使用了糟糕的索引，或者以错误的顺序执行了连接。

### SQL ordered by Elapsed Time

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- % Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100

Elapsed Time (s)	CPU Time (s)	Executions	Elap per Exec (s)	% Total DB Time	SQL ID	SQL Module	SQL Text
178	174	1,191	0.15	41.77	9b8k7vch3vch3h	OEM SystemPool	BEIGN END_NOTIFICATION_QUEUE_R...
22	13	499	0.04	5.09	9qgch3vch3v3h		DECLARE job BRARY_INTEGER = ...
12	12	14,296	0.00	2.97	9b75vch3vch3h	OEM SystemPool	begin MGMT_JOB_ENGINE_get_sche...
9	9	10	0.88	2.06	9b8k7vch3vch3h		insert into wntd_sgn_target_ad...
8	2	120	0.07	1.97	9b8k7vch3vch3h	emagent.exe	/* OracleOEM */ DECLARE UJ...
5	0	36	0.13	1.07	9b8k7vch3vch3h	perl.exe	select log_mode from v\$databas...
4	1	120	0.04	0.99	9b8k7vch3vch3h	emagent.exe	SELECT (100 - sum(percent_spc...
3	3	796	0.00	0.72	9b8k7vch3vch3h	OEM SystemPool	INSERT INTO MGMT_METRICS_RAANC...
3	2	361	0.01	0.69	9b8k7vch3vch3h		CALL MGMT_ADMIN_DATA_EVALUATE...
3	3	599	0.00	0.62	9b8k7vch3vch3h	OEM SystemPool	BEIGN EM_PING_RECORD_BATCH_HEA...

#### SQL ordered by CPU Time

- Resources reported by RUSQL code includes the resources used by all SQL statements collected by the code
- Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100

CPU Time (s)	Elapsed Time (s)	Executions	CPU per Exec (s)	% Total DB Time	SQL Id	SQL Module	SQL Text
174	170	1,181	0.15	41.77	2b064b2a1wb74n	OEM SystemPool	BEGIN EMD_NOTIFICATION_QUEUE_R...
13	22	499	0.03	5.09	8a9c1f0a3a3a		DECLARE job_owner# VARCHAR2(100)...
12	12	14,266	0.00	2.87	4b15f0e3c7f1f1b	OEM SystemPool	begin MGMT_JOB_ENGINE.get_sche...
9	9	10	0.87	2.06	3a5a5a5a5a5a5a		insert into v\$tbl_name_target_val...
3	3	796	0.00	0.72	0b1a7a5a5a5a5a	OEM SystemPool	INSERT INTO MGMT_METRICS_RAWC...
3	3	599	0.00	0.62	5a5a5a5a5a5a5a	OEM SystemPool	BEGIN EMD_PING_RECORD_BATCH_HEA...
2	3	361	0.01	0.69	4b15f0e3c7f1f1b		CALL MGMT_ADMIN_DATA_EVALUATE...
2	8	120	0.02	1.97	9a5a5a5a5a5a5a	emagent.exe	Oracle OEM 11 DECLARE U...
1	1	14,266	0.00	0.33	8a9c1f0a3a3a3a	OEM SystemPool	SELECT JOB_ID, EXECUTION_ID...
1	1	240	0.01	0.31	3a5a5a5a5a5a5a	OEM SystemPool	INSERT INTO mgmt_ha_pshvCOLLE...
0	5	36	0.01	1.07	1b1a7a5a5a5a5a	perl.exe	select job_name from v\$jobtab...

#### SQL ordered by Reads

- Total Disk Reads: 3,202
- Captured SQL account for 64.7% of Total

Physical Reads	Executions	Reads per Exec	%Total	CPU Time (s)	Elapsed Time (s)	SQL Id	SQL Module	SQL Text
454	20	22.70	14.18	0.74	1.02	839a5a5a5a5a5a		select OBJECT_ID, CLOB, RUNT...
434	1	434.00	13.55	0.50	1.01	9a5a5a5a5a5a5a		select s.owner#, s.obj, s.desc...
163	499	0.33	5.09	13.18	21.62	8a9c1f0a3a3a3a		DECLARE job_owner# VARCHAR2(100)...
107	361	0.30	3.34	2.30	2.95	4b15f0e3c7f1f1b		CALL MGMT_ADMIN_DATA_EVALUATE...
91	2,518	0.04	2.84	0.71	0.72	5a5a5a5a5a5a5a		insert into sys_wrtb_optidat_h...
55	48	1.15	1.72	0.08	0.17	0b1a7a5a5a5a5a		insert into sys_wrtb_optidat_h...
49	1,180	0.04	1.53	0.33	0.53	9a5a5a5a5a5a5a		select owner#, name, namespace...
36	2,110	0.02	1.09	1.22	1.88	8a9c1f0a3a3a3a		insert into sys_wrtb_optidat_h...
33	274	0.12	1.03	0.11	0.17	5a5a5a5a5a5a5a		insert into sys_wrtb_optidat_h...
26	8	3.25	0.81	0.02	0.04	3a5a5a5a5a5a5a		insert into sys_wrtb_optidat_h...

#### SQL ordered by Parse Calls

- Total Parse Calls: 64,938
- Captured SQL account for 56.4% of Total

Parse Calls	Executions	% Total Parse	SQL Id	SQL Module	SQL Text
2,150	2,147	3.31	0b6b2a1wb74n		select privilege#, level from ...
1,564	1,564	2.41	8bd38umbhbpqsz		select parttype, partcnt, pa...
1,556	1,556	2.40	8129566qyxz21	OEM SystemPool	SELECT INSTANTIABLE, supertyp...
1,209	3,048	1.86	3c1kubcdmnpq		update sys.col_usage\$ set eq...
1,209	1,209	1.86	b2amnm5c6r51n		lock table sys.col_usage\$ in e...
1,181	1,181	1.82	2b064b2a1wb74n	OEM SystemPool	BEGIN EMD_NOTIFICATION_QUEUE_R...
1,180	1,180	1.82	8swvpbbrbm372		select order#, columns, types ...
1,180	1,180	1.82	eqqv56fmu63x		select owner#, name, namespace...
1,035	1,035	1.59	359f5vnmnmshs		lock table sys.mon_mods\$ in ex...
1,035	1,035	1.59	q06c285rmgsw		update sys.mon_mods\$ set inser...
994	1,533	1.53	24dcd83u3rj6k		SELECT COUNT(*) FROM MGMT_PARA...
958	314	1.48	53bf0d8f9bs8		insert into sys.col_usage\$ val...
828	828	1.28	27xadvfrq5q14		select u.name, o.name, o.nam...
710	710	1.09	0k8522rmzqz8k		select privilege# from sysauth...

#### SQL ordered by Executions

- Total Executions: 221,838
- Captured SQL account for 59.6% of Total

Executions	Rows Processed	Rows per Exec	CPU per Exec (s)	Seg per Exec (s)	SQL Id	SQL Module	SQL Text
14,266	1	0.00	0.00	0.00	8a9c1f0a3a3a3a	OEM SystemPool	SELECT JOB_ID, EXECUTION_ID...
14,266	14,266	1.00	0.00	0.00	4b15f0e3c7f1f1b	OEM SystemPool	begin MGMT_JOB_ENGINE.get_sche...
13,340	13,340	1.00	0.00	0.00	0b1a7a5a5a5a5a	SQLPlus	SELECT SYS_EXTRACT_UTCTOYSTIME...
8,925	8,925	1.00	0.00	0.00	9a5a5a5a5a5a5a	OEM SystemPool	UPDATE MGMT_CURRENT_METRICS SE...
7,735	7,451	0.96	0.00	0.00	9a5a5a5a5a5a5a	OEM SystemPool	select P# rule "1bucket_cst"...
496	496	1.05	0.00	0.00	0b1a7a5a5a5a5a		select job, m2list_sche...
5,870	5,870	1.00	0.00	0.00	9a5a5a5a5a5a5a		select count(*) from sys.job\$
3,040	2,734	0.90	0.00	0.00	3c1kubcdmnpq		update sys.col_usage\$ set eq...
2,518	2,518	1.00	0.00	0.00	33b0f5nmnmshs		update hist_header\$ set bucket_c...
2,518	2,518	1.00	0.00	0.00	5a5a5a5a5a5a5a		insert into sys_wrtb_optidat_h...

在10g中，v\$sql显示使用相同语句的多个用户的SQL，V\$SQL\_PLAN\_STATISTICS\_ALL显示连接计划和统计数据，V\$SQL\_PLAN\_STATISTICS显示

## 六、实例活动统计数据

比较在内存中和磁盘中的排序量，如果磁盘排序太高就需要增加PGA\_AGGREGATE\_TARGET(或者旧版本中增大SORT\_AREA\_SIZE)

如果磁盘的读操作较高，表明可能执行了全表扫描，如果目前存在大量的较大的对较大表的全表扫描，就应当评估最常用的查询

并通过增加索引来提高效率。大量的非一致性读操作意味着使用了过多的索引或者使用了非选择性索引。如果脏读缓冲区数量高于

所请求的空闲缓冲区的数量（超过5%），那么说明DB\_CACHE\_SIZE太小，或者没有建立足够多的检查点。如果叶节点的分裂数量很高

可以考虑重建已增长或已经碎化的索引。

consistent gets:没有使用select for update子句的查询在缓冲中访问的数据块数量，这个数量加上DB BLOCK GETS统计信息的值

就是逻辑读操作总数

DB BLOCK GETS:使用了INSERT UPDATE DELETE OR SELECT FOR UPDATE语句在缓存中访问的数据块数量。

PHYSICAL READS:没有从缓存中度取得数据量。可以从磁盘，操作系统缓存或者磁盘缓存中读取，以满足

SELECT, SELECT FOR UPDATE,

INSERT, UPDATE, DELETE语句

LOGICAL READS=CONSISTENT GETS+DB BLOCK GETS

缓存命中率HIT RATIO=(LOGICAL READS- PHYSICAL READS)/LOGICAL READS \*100%



$$= (\text{CONSISTENT GETS} + \text{DB BLOCK GETS} - \text{PHYSICAL READS}) / (\text{CONSISTENT GETS} + \text{DB BLOCK GETS}) * 100\%$$

缓存命中率应该高于95%，否则需要增加DB\_CACHE\_SIZE。

DIRTY BUFFERS INSPECTED: 从LRU列表中清除掉的脏读（经过修改的）数据缓冲区的数量，如果此值超过0，可以考虑增加DB\_WR进程。

ENQUEUE TIMEOUTS: 请求入列的次数（锁定），以及所请求的特定队列不可用的次数。如果这个统计信息大于0，就需要调查锁定问题。

FREE BUFFER INSPECTED: 由于是脏读数据、被固定或者正忙等原因儿跳过的缓冲区数量。如果数量很大的话就说明缓冲区缓存太小了。

PARSE COUNT: 一条SQL语句被解析的次数。

RECURSIVE CALLS: 数据库中递归调用的数量。如果某个进程中的递归调用数量大于4，就应当检查数据字典缓存的命中率，

以及是否有表或者索引的范围过大。除非使用了大量PL/SQL，否则在用户调用中，递归调用所占的比例应该低于10%。

REDO SIZE: 写入日志中，以字节为单位的重做信息的数量。该信息将有助于确定重做日志的大小。

SORTS(DISK): 磁盘排序的数量。磁盘排序除以内存排序数量不应该高于5%。否则需要调整

SORT\_AREA\_SIZE, PGA\_AGGREGATE\_TARGET的大小

注意：SORT\_AREA\_SIZE分配的内存是面向每个用户的，PGA\_AGGREGATE\_TARGET分配的内存是面向所有会话的。

SORTS(MEMORY): 在内存中排序的数量。

SORTS(ROWS): 参加排序的数据行的数量。

TABLE FETCH BY ROWID: 通过访问ROWID访问的数据行的数量。该值很高通常意味着就获取数据的操作而言，应用程序调整的不错。

TABLE FETCH CONTINUED ROW: 获取的数据行的数量，可以是链化数据行，也可以是迁移的数据行。

## 七、表空间和文件 I/O 统计数据

对于带缓存的磁盘I/O时间通常少于1ms。

在init.ora文件中可以设置参数DB\_FILE\_MULTIBLOCK\_READ\_COUNT有助于磁盘读取时间，该参数控制在全表扫描时一次I/O中读入的

数据块数量，这将减少扫描一张表所需的I/O数量，从而提高全表扫描的性能。但是，设置该参数的结果是优化器可能会执行更多的

全表扫描，所以需要将OPTIMIZER\_INDEX\_COST\_ADJ设为一个值，例如10，来消除这个问题，并且驱动索引的使用。

数据字典和库缓存的统计数据

如果报表中PCT MISS值很高，你应当提高应用程序中游标的共享程度或者增加共享池的尺寸。

AWR报表和STATSPACK输出结果中首先需要查看的10项内容

- 1) 首要的5个等待时间；
- 2) 负载简档；
- 3) 实例效率和命中率；
- 4) 等待事件；
- 5) 阻塞等待；
- 6) 首要的SQL；
- 7) 实例活动；
- 8) 文件I/O和段统计数据；
- 9) 内存分配；
- 10) 缓冲区等待；