



恩墨科技 成就所托

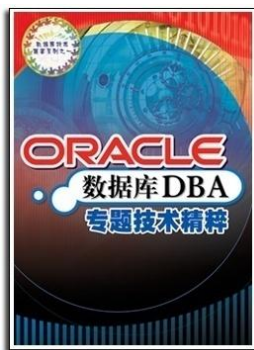
WWW.ENMOU.COM

入微 – AWR报告解读与数据库技术深入解析

- 盖国强 (eygle)
北京恩墨科技
- Mobile:13911812803
- MSN: eygle@hotmail.com
- Site : www.eygle.com
- Mail: eygle@eygle.com

Who am I

- 10+ 年 Oracle数据库经验
- 北京恩墨科技有限公司 创始人
- ITPUB论坛超级版主
- Oracle ACE 总监
- 博客站点: www.eygle.com
- 公司站点: www.enmou.com
- 成长于网络、回馈于网络



2004



2005



2006



2007



2008



2009

首页

技术基础

备份恢复

SQL优化

诊断案例

BLOG

留言板

服务

生活

网摘

阅读

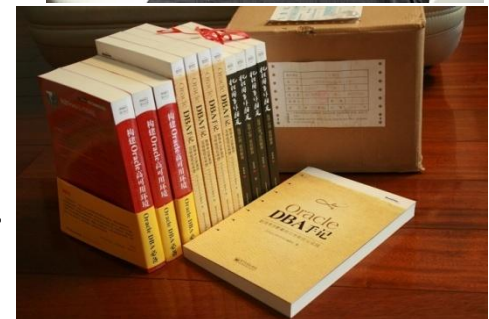
下载

eRss

English

What's Acoug?

- Acoug是一个平台、一个理想
 - 我们为讲者提供舞台
 - 我们为听者提供板凳
 - 我们为写者提供稿纸
 - 我们将来希望为讲者提供机票
 - 我们将来希望为写者提供出版
 - 我们将来希望为听者提供午餐
 - 我们已经可以为大家提供礼物!



Thanks Our Sponsor

- Oracle Univercity – Oracle大学

– Learn Oracle from Oracle



- BroadView – 博文视点

– 制作精品图书 和向上的心一起飞翔



- OTN Community – OTN社区

– 交流使经验增值！



- Eygle.Com + dbform.Com

– Friend Life Oracle



Thanks Our Speaker

- Eygle Gai

– 盖国强



- Kamus Zhang

– 张乐奕



- Dbsnake Cui

– 崔华



Case Study: row cache objects

- Row cache objects 导致数据库挂起

DB Name	DB Id	Instance	Inst num	Release	RAC	Host
NOA	1251528420	noa 2	2	10.2.0.4.0	YES	db2

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	2895	01-Mar-10 20:00:42	46	1.9
End Snap:	2896	01-Mar-10 21:00:05	47	3.2
Elapsed:		59.38 (mins)		
DB Time:		88.34 (mins)		

Top 5 Timed Events

Event	Waits	Time(s)	Avg Wait(ms)	% Total Call Time	Wait Class
latch: row cache objects	80,797	1,782	22	33.6	Concurrency
row cache lock	550,397	1,161	2	21.9	Concurrency
CPU time		830		15.7	
cursor: pin S wait on X	81,249	795	10	15.0	Concurrency
db file sequential read	159,584	705	4	13.3	User I/O

诊断 - 对比从差异入手



Dictionary Cache Stats

Dictionary Cache Stats

- "Pct Misses" should be very low (< 2% in most cases)
- "Final Usage" is the number of cache entries being used

- "Pct Misses" should be very low (< 2% in most cases)
- "Final Usage" is the number of cache entries being used

Cache	Get Requests	Pct Miss	Cache	Get Requests	Pct Miss	Scan Reqs	Pct Miss	Mod Reqs	Final Usage
dc_awr_control	65	1.54	dc_awr_control	60	1.67	0		1	1
dc_database_links	428	0.00	dc_database_links	125	0.80	0		0	0
dc_global_oids	566	0.00	dc_global_oids	65	10.77				
dc_histogram_data	1,881	0.00	dc_histogram_data	4,554	29.23				
dc_histogram_defs	10,555	0.00	dc_histogram_defs	12,266	73.16				
dc_object_ids	2,186	3.20	dc_object_ids	5,880	15.12				
dc_objects	1,036	6.08	dc_objects	2,465	28.03				
dc_profiles	284	0.00	dc_profiles	85	0.00				
dc_rollback_segments	6,233	0.00	dc_rollback_segments	2,205	0.00				
dc_segments	6,068	6.21	dc_segments	630,622	84.78				
dc_sequences	17,039	1.30	dc_sequences	2,240	1.83				
dc_tablespace_quotas	3,523	0.57	dc_tablespace_quotas	1,130	1.42				
dc_tablespaces	18,181	0.00	dc_tablespaces	5,146	0.35				
dc_usernames	1,529	0.00	dc_usernames	1,267	1.03				
dc_users	31,087	0.00	dc_users	9,951	0.08				
outstanding_alerts	286	87.41	outstanding_alerts	296	86.82	0		29	0

Dictionary Cache Stats (RAC)

Cache	GES Requests	GES Conflicts	GES Releases
dc_awr_control	3	1	0
dc_database_links	1	0	2
dc_global_oids	7	0	32
dc_histogram_defs	8,974	0	9,894
dc_object_ids	890	0	1,421
dc_objects	693	0	1,002
dc_segments	536,921	35	512,367
dc_sequences	4,474	38	4
dc_tablespace_quotas	2,260	10	1
dc_tablespaces	18	0	27
dc_usernames	13	0	19
dc_users	7	0	30
outstanding_alerts	572	241	4

诊断 - 对比从差异入手



- Load Profile中的每一个项目都值得仔细分析
– AWR对比报告是非常有益的分析手段

Load Profile

	Per Second	Per Transaction
Redo size:	794,059.85	771,519.55
Logical reads:	3,722.10	3,616.45
Block changes:	1,295.90	1,259.11
Physical reads:	78.43	76.21
Physical writes:	38.62	37.52
User calls:	7.57	7.35
Parses:	155.28	150.88
Hard parses:	0.18	0.18
Sorts:	1.87	1.82
Logons:	0.05	0.05
Executes:	164.18	159.52
Transactions:	1.03	

% Blocks changed per Read:	34.82	Recursive Call %:	99.79
Rollback per transaction %:	1.83	Rows per Sort:	180.18

Load Profile

	Per Second	Per Transaction
Redo size:	1,719,501.07	231,616.85
Logical reads:	6,435.97	866.93
Block changes:	3,377.19	454.91
Physical reads:	75.50	10.17
Physical writes:	112.35	15.13
User calls:	16.42	2.21
Parses:	11.01	1.48
Hard parses:	0.04	0.01
Sorts:	1.47	0.20
Logons:	0.10	0.01
Executes:	19.34	2.61
Transactions:	7.42	

% Blocks changed per Read:	52.47	Recursive Call %:	89.99
Rollback per transaction %:	0.27	Rows per Sort:	331.90

诊断 - 对比从差异入手



• 从时间模型分析消耗

Time Model Statistics

- Total time in database user-calls (DB Time): 1701.6s
- Statistics including the word "background" measure background
- Ordered by % or DB time desc, Statistic name

Statistic Name	Time (s)	% of DB Time
sql execute elapsed time	1,680.22	98.74
DB CPU	756.60	44.46
sequence load elapsed time	11.35	0.67
parse time elapsed	10.09	0.59
hard parse elapsed time	8.69	0.51
connection management call elapsed time	0.72	0.04
PL/SQL execution elapsed time	0.61	0.04
PL/SQL compilation elapsed time	0.02	0.00
repeated bind elapsed time	0.00	0.00
DB time	1,701.62	
background elapsed time	161.16	
background cpu time	91.91	

Time Model Statistics

- Total time in database user-calls (DB Time): 5300.2s
- Statistics including the word "background" measure background
- Ordered by % or DB time desc, Statistic name

Statistic Name	Time (s)	% of DB Time
parse time elapsed	3,531.99	66.64
sql execute elapsed time	3,496.02	65.96
hard parse elapsed time	3,190.85	60.20
DB CPU	830.09	15.66
failed parse elapsed time	379.54	7.16
connection management call elapsed time	179.62	3.39
sequence load elapsed time	1.67	0.03
PL/SQL execution elapsed time	0.23	0.00
PL/SQL compilation elapsed time	0.15	0.00
repeated bind elapsed time	0.05	0.00
hard parse (sharing criteria) elapsed time	0.04	0.00
DB time	5,300.24	
background elapsed time	1,565.53	
background cpu time	213.64	

分析-从疑问逐层深入

• SQL解析与执行

SQL ordered by Executions

- Total Executions: 584,963
- Captured SQL account for 95.6% of Total

Executions	Rows Processed	Rows per Exec	CPU per Exec (s)	Elap per Exec (s)	SQL Id	SQL Module	SQL Text
535,439	535,441	1.00	0.00	0.00	2ym6hhaq30r73		select type#, blocks, extents,...
9,863	8,304	0.84	0.00	0.00	96g93hntrzjtr		select /*+ rule */ bucket_cnt,...
2,691	2,691	1.00	0.00	0.15	7h35uxf5uhmm1	SQL*Plus	select sysdate from dual
1,242	1,431	1.15	0.00	0.00	53saa2zkr6wc3		select intcol#, nvl(pos#, 0), ...

SQL ordered by Parse Calls

[2ym6hhaq30r73](#) select type#, blocks, extents, minexts, maxexts, extsize, extpct, user#, iniexts, NVL(lists, 65535), NVL(groups, 65535), cachehint, hwmincr, NVL(spare1, 0), NVL(scanhint, 0) from seg\$ where ts#=:1 and file#=:2 and block#=:3

Parse Calls	Executions	% Total Parses	SQL Id	SQL Module	SQL Text
535,440	535,439	96.78	2ym6hhaq30r73		select type#, blocks, extents,...
2,691	2,691	0.49	7h35uxf5uhmm1	SQL*Plus	select sysdate from dual
1,066	1,066	0.19	bsa0wjftg3uw		select file# from file\$ where ...

分析-从疑问逐层深入

```
select type#,blocks,extents,minexts,maxexts,extsize,extpct,user#,iniexts,  
       NUL(lists,65535),NUL(groups,65535),cachehint,hwmincr, NUL(spare1,0),  
       NUL(scanhint,0)  
from  
seg$ where ts#=:1 and file#=:2 and block#=:3
```

call	count	cpu	elapsed	disk	query	current	rows
Parse	8	0.00	0.00	0	0	0	0
Execute	8	0.01	0.02	0	0	0	0
Fetch	8	0.00	0.00	0	23	0	7
total	24	0.01	0.02	0	23	0	7

Misses in library cache during parse: 1

Misses in library cache during execute: 1

Optimizer mode: CHOOSE

Parsing user id: SYS (recursive depth: 2)

Rows	Row Source Operation
1	TABLE ACCESS CLUSTER SEG\$ (cr=3 pr=0 pw=0 time=100 us)
1	INDEX UNIQUE SCAN I_FILE#_BLOCK# (cr=2 pr=0 pw=0 time=43 us)(object id 9)

分析-从疑问逐层深入

- What's simulator

Latch Activity

- "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	Wait Time (s)	NoWait Requests	Pct NoWait Miss
shared pool	2,057,445	3.10	0.11	6	0	
shared pool sim alloc	1	0.00		0	0	
shared pool simulator	413,655,808	0.00	0.00	0	0	
simulator hash latch	795,224	0.00		0	0	

结论与方法

- 海量分区定义导致大量递归SQL调用
- Shared Pool的老化与维护产生竞争
- 望闻问切为诊断之道
- 细致入微为定位之本

Q&A



恩墨科技 成就所托

WWW.ENMOU.COM

