开Oracle调优鹰眼，深入理解AWR性能报告- 第二讲

刘相兵(Maclean Liu)
liu.maclean@gmail.com

ORA-ALLSTARS Exadata用户组QQ群:23549328

About Me

ORACLE Certified Master
Oracle Database 10g Administrator

ORACLE Certified Master
Oracle Database 11g Administrator

- **Email:liu.maclean@gmail.com**
- **Blog:www.askmaclean.com**
- **Oracle Certified Database Administrator Master 10g and 11g**
- **Over 7 years experience with Oracle DBA technology**
- **Over 8 years experience with Linux technology**
- **Member Independent Oracle Users Group**
- **Member All China Users Group**
- **Presents for advanced Oracle topics: RAC, DataGuard, Performance Tuning and Oracle Internal.**

IOUG
independent oracle users group
For the Complete Technology & Database Professional

ACOUG
All China Oracle User Group
MEMBER

About Me

大侠请循序渐进，别漏了第一讲：http://t.askmaclean.com/thread-2086-1-1.html

Hawk Eyes 看AWR的鹰眼= 基础理论夯实+看过500份以上AWR

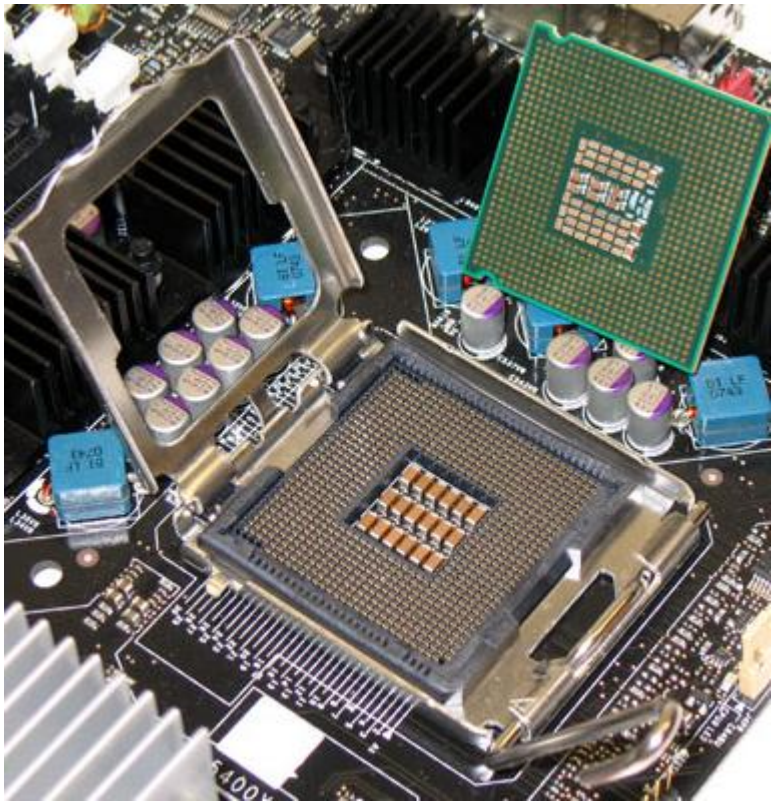
预备知识-CPU Socket

Host CPU (CPUs: 16 Cores: 8 Sockets: 2)

Load Average Begin	Load Average End	%User	%System	%WIO	%Idle
5.10	7.43	26.9	3.0	0.0	67.7

Sockets ➔ 物理CPU数目

USER_TIME	1,424,443	
LOAD	5	7
PHYSICAL_MEMORY_BYTES	75,804,418,048	
NUM_CPUS	16	
NUM_CPU_CORES	8	
NUM_CPU_SOCKETS	2	
GLOBAL_RECEIVE_SIZE_MAX	4,194,304	
GLOBAL_SEND_SIZE_MAX	2,097,152	
TCP_RECEIVE_SIZE_DEFAULT	87,380	
TCP_RECEIVE_SIZE_MAX	4,194,304	
TCP_RECEIVE_SIZE_MIN	4,096	
TCP_SEND_SIZE_DEFAULT	16,384	
TCP_SEND_SIZE_MAX	4,194,304	
TCP_SEND_SIZE_MIN	4,096	



预备知识-CPU Cores

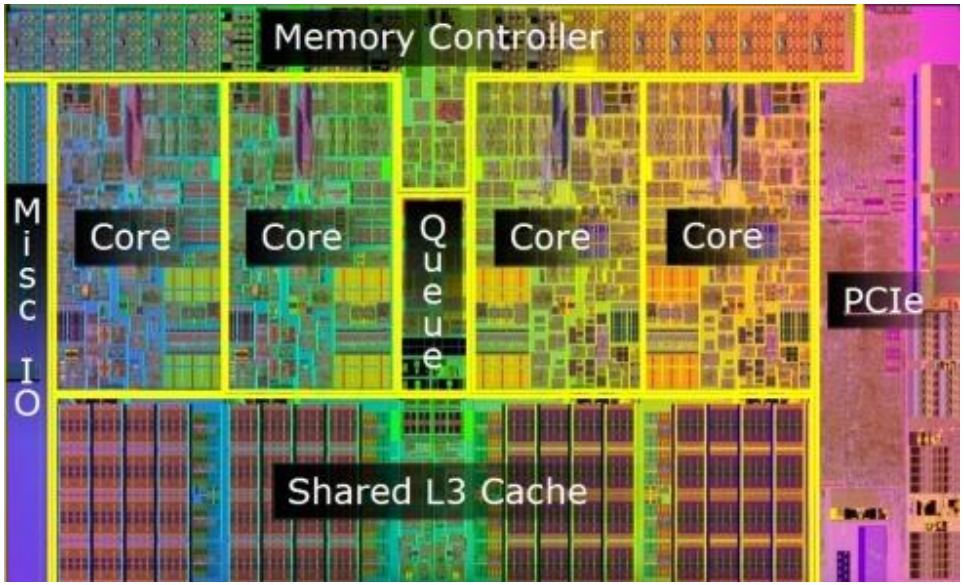
Host CPU (CPUs: 16 Cores: 8 Sockets: 2)

Load Average Begin	Load Average End	%User	%System	%WIO	
5.10	7.43	26.9	3.0	0.0	

Cores ➔ CPU核数

一个物理CPU可以对应多个core

NICE_TIME	0	
SYS_TIME	157,135	
USER_TIME	1,424,443	
LOAD	5	7
PHYSICAL_MEMORY_BYTES	75,804,418,048	
NUM_CPUS	16	
NUM_CPU_CORES	8	
NUM_CPU_SOCKETS	2	
GLOBAL_RECEIVE_SIZE_MAX	4,194,304	
GLOBAL_SEND_SIZE_MAX	2,097,152	
TCP_RECEIVE_SIZE_DEFAULT	87,380	
TCP_RECEIVE_SIZE_MAX	4,194,304	
TCP_RECEIVE_SIZE_MIN	4,096	
TCP_SEND_SIZE_DEFAULT	16,384	



预备知识-CPUs

Host CPU (CPUs: 64 Cores: 32 Sockets:)

Load Average Begin	Load Average End	%User	%System	
19.32	21.48	24.0	5.0	

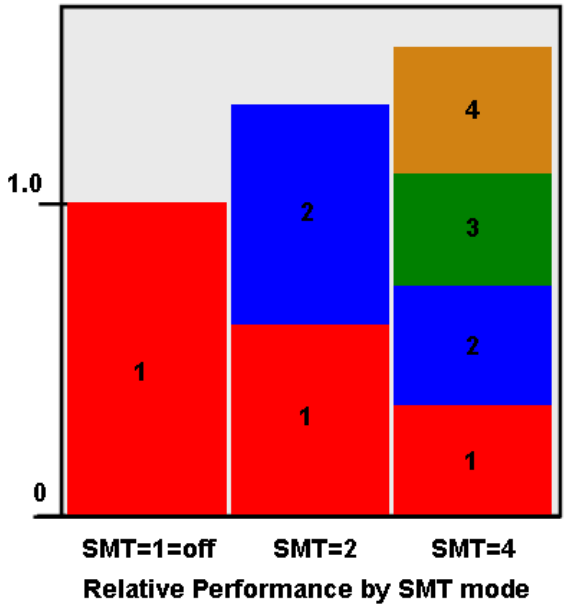
这里的CPU数是指逻辑CPU数(Number of logical CPUs available (includes hardware threads if hardware threading is turned on))

当硬件线程技术打开时，例如：
POWER SMT(Simultaneous Multi-Thread)
Intel HT(Hyper Thread)

此时一个CPU core被视作2个或更多个逻辑CPU

LOAD	19	21
OS_CPU_WAIT_TIME	34,559,400	
VM_IN_BYTES	2,799,747,072	
VM_OUT_BYTES	3,876,667,392	
PHYSICAL_MEMORY_BYTES	266,556,407,808	
NUM_CPUS	64	
NUM_CPU_CORES	32	
NUM_LCPUS	64	
NUM_VCPUS	32	
GLOBAL_RECEIVE_SIZE_MAX	4,194,304	
GLOBAL_SEND_SIZE_MAX	4,194,304	

Simultaneous Multi-Threading (SMT)



NUM_LCPUS仅在11.2.0.2之后的报告中出现

Host CPU

Host CPU (CPUs: 64 Cores: 32 Sockets:)

Load Average Begin	Load Average End	%User	%System	%WIO	%Idle
19.32	21.48	24.0	5.0	17.5	71.0

“Load Average” begin/end值代表CPU的大致运行队列大小。上图中快照开始到结束，平均 CPU负载增加了。

%User+%System=> 总的CPU使用率，在这里是29.0%

Elapsed Time * NUM_CPUS * CPU utilization= 299.70 (mins) * 64 * 29%= 5562.4 mins=Busy Time
与《**Operating System Statistics**》中的**LOAD**相呼应

IDLE_TIME	81,844,586	
IOWAIT_TIME	20,143,489	
SYS_TIME	5,802,535	
USER_TIME	27,646,183	
LOAD	19	21
OS_CPU_WAIT_TIME	34,559,400	
VM_IN_BYTES	2,799,747,072	
VM_OUT_BYTES	3,876,667,392	
PHYSICAL_MEMORY_BYTES	266,556,407,808	
NUM_CPUS	64	

Instance CPU

Instance CPU

%Total CPU	%Busy CPU	%DB time waiting for CPU (Resource Manager)
22.4	77.1	0.0

%Total CPU,该实例所使用的CPU占总CPU的比例 ➔ % of total CPU for Instance

%Busy CPU，该实例所使用的Cpu占总的被使用CPU的比例 ➔ % of busy CPU for Instance

例如共4个逻辑CPU，其中3个被完全使用，3个中的1个完全被该实例使用，则%Total CPU= ¼ =25%，而%Busy CPU= 1/3= 33%

当CPU高时一般看%Busy CPU可以确定CPU到底是否是本实例消耗的，还是主机上其他程序

% of busy CPU for Instance

Instance CPU

%Total CPU	%Busy CPU	%DB time waiting for CPU (Resource Manager)
22.4	77.1	0.0

% of busy CPU for Instance= （DB CPU+ background cpu time) / (BUSY_TIME /100)= (250,777.19 + 7,195.70)/ (33,448,718/100)= 77.1%

DB CPU	250,777.19	38.22
parse time elapsed	31,931.81	4.87
hard parse elapsed time	23,704.43	3.61
sequence load elapsed time	4,811.47	0.73
PL/SQL compilation elapsed time	4,071.82	0.62
connection management call elapsed time	2,286.88	0.35
failed parse elapsed time	798.64	0.12
PL/SQL execution elapsed time	543.67	0.08
hard parse (sharing criteria) elapsed time	409.33	0.06
hard parse (bind mismatch) elapsed time	260.95	0.04
repeated bind elapsed time	35.47	0.01
RMAN cpu time (backup/restore)	0.41	0.00
DB time	656,135.87	
background elapsed time	25,023.07	
background cpu time	7,195.70	

AVG_IOWAIT_TIME	313,955	
AVG_SYS_TIME	89,927	
AVG_USER_TIME	431,162	
BUSY_TIME	33,448,718	
IDLE_TIME	81,844,586	
IOWAIT_TIME	20,143,489	
SYS_TIME	5,802,535	
USER_TIME	27,646,183	
LOAD	19	21
OS_CPU_WAIT_TIME	34,559,400	
VM_IN_BYTES	2,799,747,072	

% of total CPU for Instance

Instance CPU

%Total CPU	%Busy CPU	%DB time waiting for CPU (Resource Manager)
22.4	77.1	0.0

% of total CPU for Instance= （DB CPU+ background cpu time) / ((BUSY_TIME+IDLE_TIME) /100)= (250,777.19 + 7,195.70)/ ((33,448,718+81,844,586/100)= 22.4%

DB CPU	250,777.19	38.22
parse time elapsed	31,931.81	4.87
hard parse elapsed time	23,704.43	3.61
sequence load elapsed time	4,811.47	0.73
PL/SQL compilation elapsed time	4,071.82	0.62
connection management call elapsed time	2,286.88	0.35
failed parse elapsed time	798.64	0.12
PL/SQL execution elapsed time	543.67	0.08
hard parse (sharing criteria) elapsed time	409.33	0.06
hard parse (bind mismatch) elapsed time	260.95	0.04
repeated bind elapsed time	35.47	0.01
RMAN cpu time (backup/restore)	0.41	0.00
DB time	656,135.87	
background elapsed time	25,023.07	
background cpu time	7,195.70	

AVG_IOWAIT_TIME	313,955	
AVG_SYS_TIME	89,927	
AVG_USER_TIME	431,162	
BUSY_TIME	33,448,718	
IDLE_TIME	81,844,586	
IOWAIT_TIME	20,143,489	
SYS_TIME	5,802,535	
USER_TIME	27,646,183	
LOAD	19	21
OS_CPU_WAIT_TIME	34,559,400	
VM_IN_BYTES	2,799,747,072	
VM_OUT_BYTES	3,876,667,392	

%DB time waiting for CPU (Resource Manager)

Instance CPU

%Total CPU	%Busy CPU		%DB time waiting for CPU (Resource Manager)
98.0	98.4		37.9

%DB time waiting for CPU (Resource Manager)= (RSRC_MGR_CPU_WAIT_TIME/100)/DB TIME= (138,776,449/100)/(61,052 * 60)= 37.88%

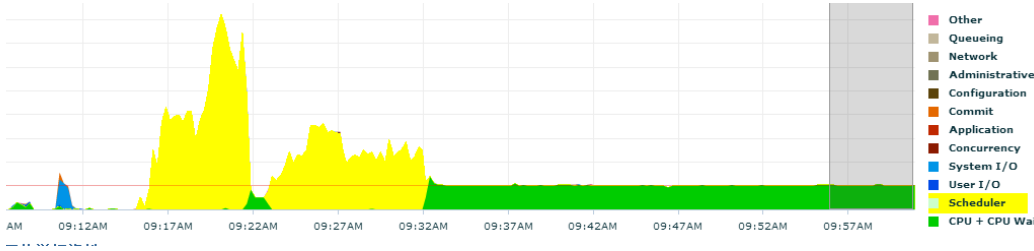
Statistic	Value	End Value
AVG_BUSY_TIME	3,763,310	
AVG_IDLE_TIME	14,645	
AVG_SYS_TIME	65,001	
AVG_USER_TIME	3,696,446	
BUSY_TIME	180,728,579	
IDLE_TIME	720,118	
SYS_TIME	3,209,167	
USER_TIME	177,519,412	
LOAD	15	82
OS_CPU_WAIT_TIME	108,832,800	
RSRC_MGR_CPU_WAIT_TIME	138,776,449	
VM_IN_BYTES	60,219,392	
PHYSICAL_MEMORY_BYTES	206,158,430,208	
NUM_CPUS	48	
NUM_CPU_CORES	48	

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	23596	13-Aug-12 23:30:45	225	1.1
End Snap:	23617	14-Aug-12 10:00:41	236	1.2
Elapsed:		629.94 (mins)		
DB Time:		61,052.69 (mins)		

真实案例resmgr:cpu quantum

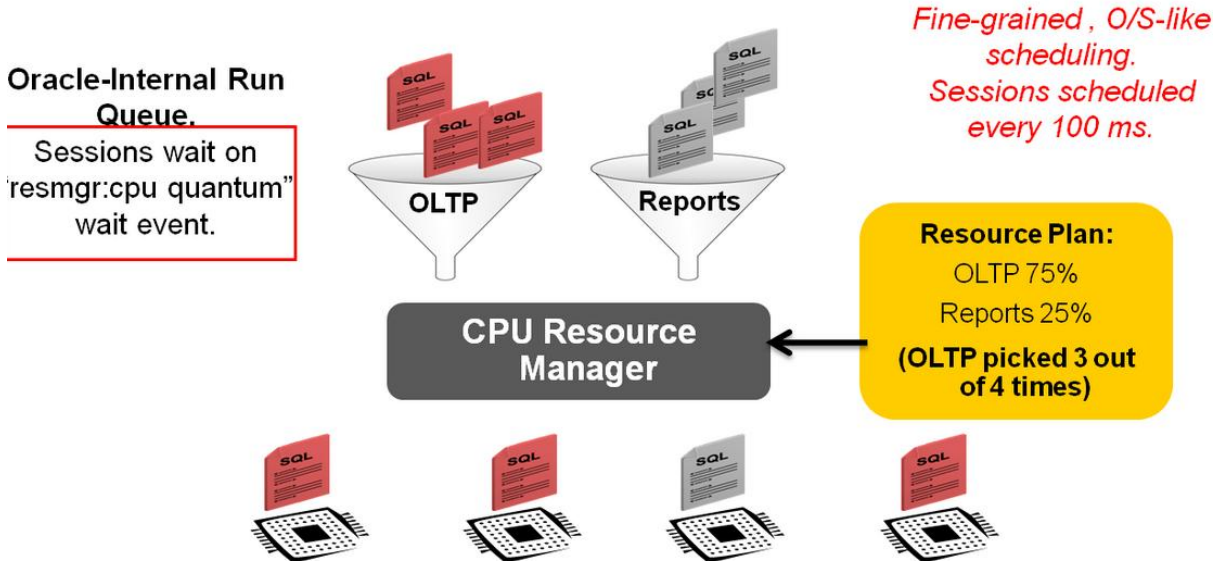
Top 5 Timed Foreground Events

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
DB CPU		1,776,521		48.50	
resmgr:cpu quantum	6,621,496	1,230,793	186	33.60	Scheduler
latch free	807,340	146,601	182	4.00	Other
buffer busy waits	38,952	12,640	325	0.35	Concurrency
direct path read	47,895	9,774	204	0.27	User I/O



resmgr:cpu quantum等待事件是当resource manager控制CPU调度时，需要控制对应进程暂时不使用CPU而进程到内部运行队列中，以保证该进程对应的consumer group(消费组)没有消耗比指定resource manager指令更多的CPU。

CPU Scheduling with Resource Manager



Statistic	Value	End Value
AVG_BUSY_TIME	3,763,310	
AVG_IDLE_TIME	14,645	
AVG_SYS_TIME	65,001	
AVG_USER_TIME	3,696,446	
BUSY_TIME	180,728,579	
IDLE_TIME	720,118	
SYS_TIME	3,209,167	
USER_TIME	177,519,412	
LOAD	15	82
OS_CPU_WAIT_TIME	108,832,800	
RSRC_MGR_CPU_WAIT_TIME	138,776,449	
VM_IN_BYTES	60,219,392	
PHYSICAL_MEMORY_BYTES	206,158,430,208	
NUM_CPUS	48	
NUM_CPU_CORES	48	
NUM_CPU_SOCKETS	24	
TCP_RECEIVE_SIZE_DEFAULT	49,152	
TCP_RECEIVE_SIZE_MAX	1,048,576	

Operating System Statistics

Operating System Statistics

- *TIME statistic values are diffed. All others display actual values. End Valu
- ordered by statistic type (CPU Use, Virtual Memory, Hardware Config), Name

Statistic	Value	End Value
AVG_BUSY_TIME	521,839	
AVG_IDLE_TIME	1,277,925	
AVG_IOWAIT_TIME	313,955	
AVG_SYS_TIME	89,927	
AVG_USER_TIME	431,162	
BUSY_TIME	33,448,718	
IDLE_TIME	81,844,586	
IOWAIT_TIME	20,143,489	
SYS_TIME	5,802,535	
USER_TIME	27,646,183	
LOAD	19	21
OS_CPU_WAIT_TIME	34,559,400	
VM_IN_BYTES	2,799,747,072	
VM_OUT_BYTES	3,876,667,392	
PHYSICAL_MEMORY_BYTES	266,556,407,808	
NUM_CPUS	64	
NUM_CPU_CORES	32	
NUM_LCPUS	64	
NUM_VCPUS	32	
GLOBAL_RECEIVE_SIZE_MAX	4,194,304	
GLOBAL_SEND_SIZE_MAX	4,194,304	
TCP_RECEIVE_SIZE_DEFAULT	16,384	
TCP_RECEIVE_SIZE_MAX	9,223,372,036,854,775,807	
TCP_RECEIVE_SIZE_MIN	4,096	
TCP_SEND_SIZE_DEFAULT	16,384	
TCP_SEND_SIZE_MAX	9,223,372,036,854,775,807	
TCP_SEND_SIZE_MIN	4,096	

数据来源于V$OSSTAT➔DBA_HIST_OSSTAT，，TIME相关的指标单位均为百分之一秒

Busy_Time=SYS_TIME+USER_TIME

AVG_BUSY_TIME= BUSY_TIME/NUM_CPUS

BUSY_TIME + IDLE_TIME = ELAPSED_TIME * CPU_COUNT=299.7*60*64=1150848s=(33,448,718+81,844,586)/100

OS_CPU_WAIT_TIME ➔进程等OS调度，cpu queuing

VM_IN_BYTES 2.6GB PGAE IN
VM_OUT_BYTES 3.6GB PGAE OUT ➔部分版本下并不准确，例如Bug 11712010 Abstract: VIRTUAL MEMORY PAGING ON 11.2.0.2 DATABASES，仅供参考

Memory Statistics

Memory Statistics

	Begin	End
Host Mem (MB):	254,208.0	254,208.0
SGA use (MB):	116,214.8	116,214.8
PGA use (MB):	23,677.9	26,301.0
% Host Mem used for SGA+PGA:	55.03	56.06

PGA Memory Advisory

- When using Auto Memory Mgmt, minimally choose a pga_aggregate_target value where Estd PGA Overalloc Count is 0

PGA Target Est (MB)	Size Factr	W/A MB Processed	Estd Extra W/A MB Read/ Written to Disk	Estd PGA Cache Hit %	Estd PGA Overalloc Count	Estd Time
2,048	0.13	5,987,899.93	5,196,327.76	54.00	486,006	638,943,511
4,096	0.25	5,987,899.93	5,180,596.41	54.00	480,272	638,044,795
8,192	0.50	5,987,899.93	5,125,470.10	54.00	471,923	634,895,486
12,288	0.75	5,987,899.93	4,919,590.77	55.00	452,260	623,133,809
16,384	1.00	5,987,899.93	24,030.10	100.00	392,636	343,455,425
19,661	1.20	5,987,899.93	24,030.10	100.00	263,608	343,455,425
22,938	1.40	5,987,899.93	24,030.10	100.00	91,205	343,455,425
26,214	1.60	5,987,899.93	24,030.10	100.00	0	343,455,425
29,491	1.80	5,987,899.93	24,030.10	100.00	0	343,455,425
32,768	2.00	5,987,899.93	24,030.10	100.00	0	343,455,425
49,152	3.00	5,987,899.93	24,030.10	100.00	0	343,455,425
65,536	4.00	5,987,899.93	24,030.10	100.00	0	343,455,425
98,304	6.00	5,987,899.93	24,030.10	100.00	0	343,455,425
131,072	8.00	5,987,899.93	24,030.10	100.00	0	343,455,425

11g以后才有的一个section，可以了解SGA、PGA、主机物理内存的大致使用情况。

如上例PGA在快照时间内从23677MB增长到26301MB, pga_aggregate_target=16GB，存在overalloc

% Host Mem used for SGA+PGA 可以大致反映本实例占用主机物理内存的情况。

注意Host Mem也可能起伏，如使用DLPAR时

Time Model Statistics

Time Model Statistics

- Total time in database user-calls (DB Time): 656135.9s
- Statistics including the word "background" measure background process time, and so do not contribute to the DB time statistic
- Ordered by % or DB time desc, Statistic name

Statistic Name	Time (s)	% of DB Time
sql execute elapsed time	593,443.33	90.45
DB CPU	250,777.19	38.22
parse time elapsed	31,931.81	4.87
hard parse elapsed time	23,704.43	3.61
sequence load elapsed time	4,811.47	0.73
PL/SQL compilation elapsed time	4,071.82	0.62
connection management call elapsed time	2,286.88	0.35
failed parse elapsed time	798.64	0.12
PL/SQL execution elapsed time	543.67	0.08
hard parse (sharing criteria) elapsed time	409.33	0.06
hard parse (bind mismatch) elapsed time	260.95	0.04
repeated bind elapsed time	35.47	0.01
RMAN cpu time (backup/restore)	0.41	0.00
DB time	656,135.87	
background elapsed time	25,023.07	
background cpu time	7,195.70	

Time Model Statistics几个特别有用的时间指标：

- parse time elapsed、hard parse elapsed time 结合起来看解析是否是主要矛盾，若是则重点是软解析还是硬解析

- sequence load elapsed time sequence序列争用是否是问题焦点

- PL/SQL compilation elapsed time PL/SQL对象编译

- 注意PL/SQL execution elapsed time 纯耗费在PL/SQL解释器上的时间。不包括花在执行和解析其包含SQL上的时间

Time Model Statistics

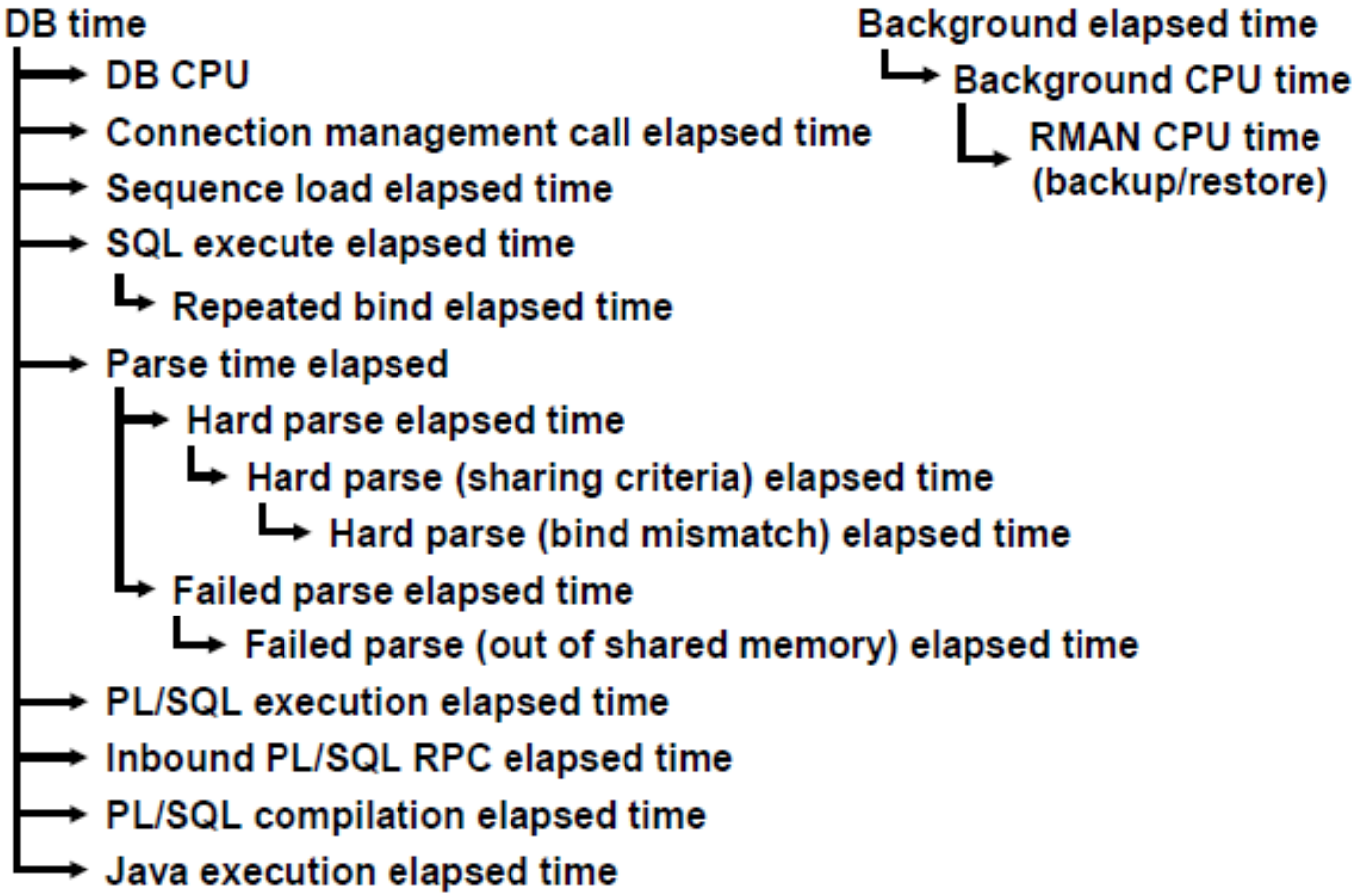
Time Model Statistics

- Total time in database user-calls (DB Time): 656135.9s
- Statistics including the word "background" measure background process time, and so do not contribute to the DB time statistic
- Ordered by % or DB time desc, Statistic name

Statistic Name	Time (s)	% of DB Time
sql execute elapsed time	593,443.33	90.45
DB CPU	250,777.19	38.22
parse time elapsed	31,931.81	4.87
hard parse elapsed time	23,704.43	3.61
sequence load elapsed time	4,811.47	0.73
PL/SQL compilation elapsed time	4,071.82	0.62
connection management call elapsed time	2,286.88	0.35
failed parse elapsed time	798.64	0.12
PL/SQL execution elapsed time	543.67	0.08
hard parse (sharing criteria) elapsed time	409.33	0.06
hard parse (bind mismatch) elapsed time	260.95	0.04
repeated bind elapsed time	35.47	0.01
RMAN cpu time (backup/restore)	0.41	0.00
DB time	656,135.87	
background elapsed time	25,023.07	
background cpu time	7,195.70	

- connection management call elapsed time 建立数据库session连接和断开
- failed parse elapsed time 解析失败，例如由于ORA-4031
- hard parse (sharing criteria) elapsed time 由于无法共享游标造成的硬解析
- hard parse (bind mismatch) elapsed time 由于bind type or bind size 不一致造成的硬解析

Time Model Statistics树形图



存在包含关系所以Time Model Statistics加起来超过100%再正常不过

Operating System Statistics - Detail

Operating System Statistics - Detail

Snap Time	Load	%busy	%user	%sys	%idle	%iowait
27-Aug 11:00:33	19.32					
27-Aug 11:30:37	16.89	25.75	20.89	4.87	74.25	17.81
27-Aug 12:00:25	14.08	21.59	17.61	3.98	78.41	13.19
27-Aug 12:30:43	17.19	23.95	20.18	3.77	76.05	13.18
27-Aug 13:00:49	18.11	24.97	20.75	4.22	75.03	13.50
27-Aug 13:30:24	20.33	26.61	21.80	4.81	73.39	19.06
27-Aug 14:00:57	22.89	35.39	29.68	5.71	64.61	17.80
27-Aug 14:30:03	26.80	36.63	30.63	6.00	63.37	20.53
27-Aug 15:00:07	21.94	32.83	27.21	5.62	67.17	19.57
27-Aug 15:30:13	21.00	31.39	25.58	5.82	68.61	20.53
27-Aug 16:00:15	21.48	31.10	25.53	5.56	68.90	19.65

每个快照对应一行记录，简易了解OS负载情况

针对CPU Spike毛刺问题有点用，也可看做大致的cpu走势

虽然替代不了nmon、osw，但胜在AWR自带。

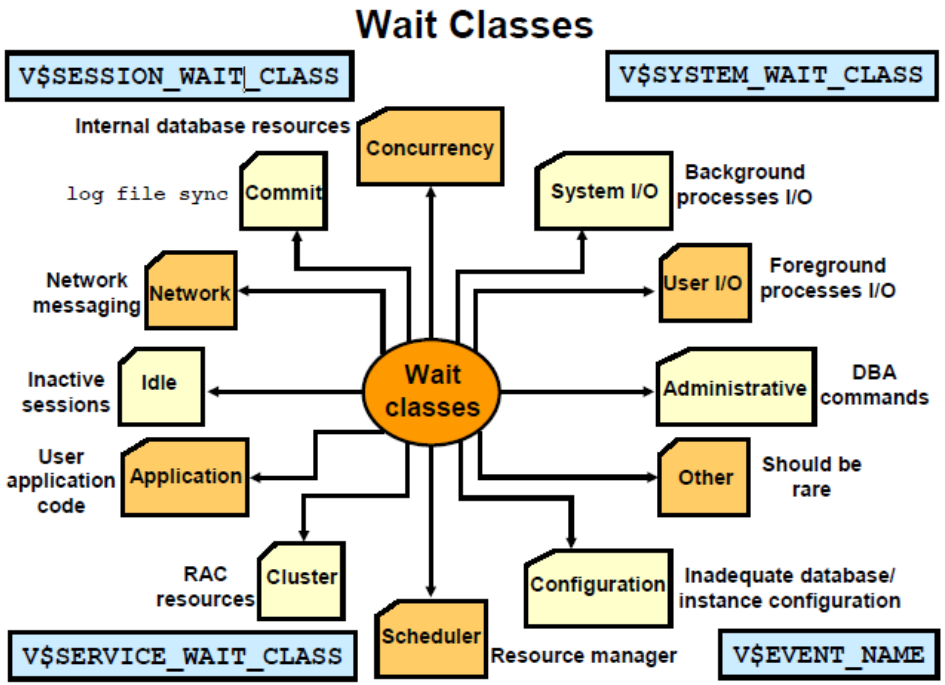
特别是对哪些没任何os性能监控的”处女环境”来说是个宝。

Wait Class

Foreground Wait Class

- s - second, ms - millisecond - 1000th of a second
- ordered by wait time desc, waits desc
- %Timeouts: value of 0 indicates value was < .5%. Value of null is truly 0
- Captured Time accounts for 96.8% of Total DB time 656,135.87 (s)
- Total FG Wait Time: 384,156.88 (s) DB CPU time: 250,777.19 (s)

Wait Class	Waits	%Time -outs	Total Wait Time (s)	Avg wait (ms)	%DB time
User I/O	74,910,691	0	299,180	4	45.60
DB CPU			250,777		38.22
Concurrency	4,766,281	0	22,564	5	3.44
Cluster	17,933,896	0	22,292	1	3.40
Other	1,458,827	82	21,411	15	3.26
Commit	4,116,154	0	11,322	3	1.73
Configuration	4,106	0	4,344	1058	0.66
Application	39,549	0	2,016	51	0.31
Network	142,066,579	0	730	0	0.11
System I/O	340,920	0	170	0	0.03
Administrative	224	100	127	568	0.02




```
SQL> select distinct wait_class from
v$event_name;

WAIT_CLASS
----------------------------------------------------------------
Concurrency
User I/O
System I/O
Administrative
Other
Configuration
Scheduler
Cluster
Application
Queueing
Idle
Network
Commit
```

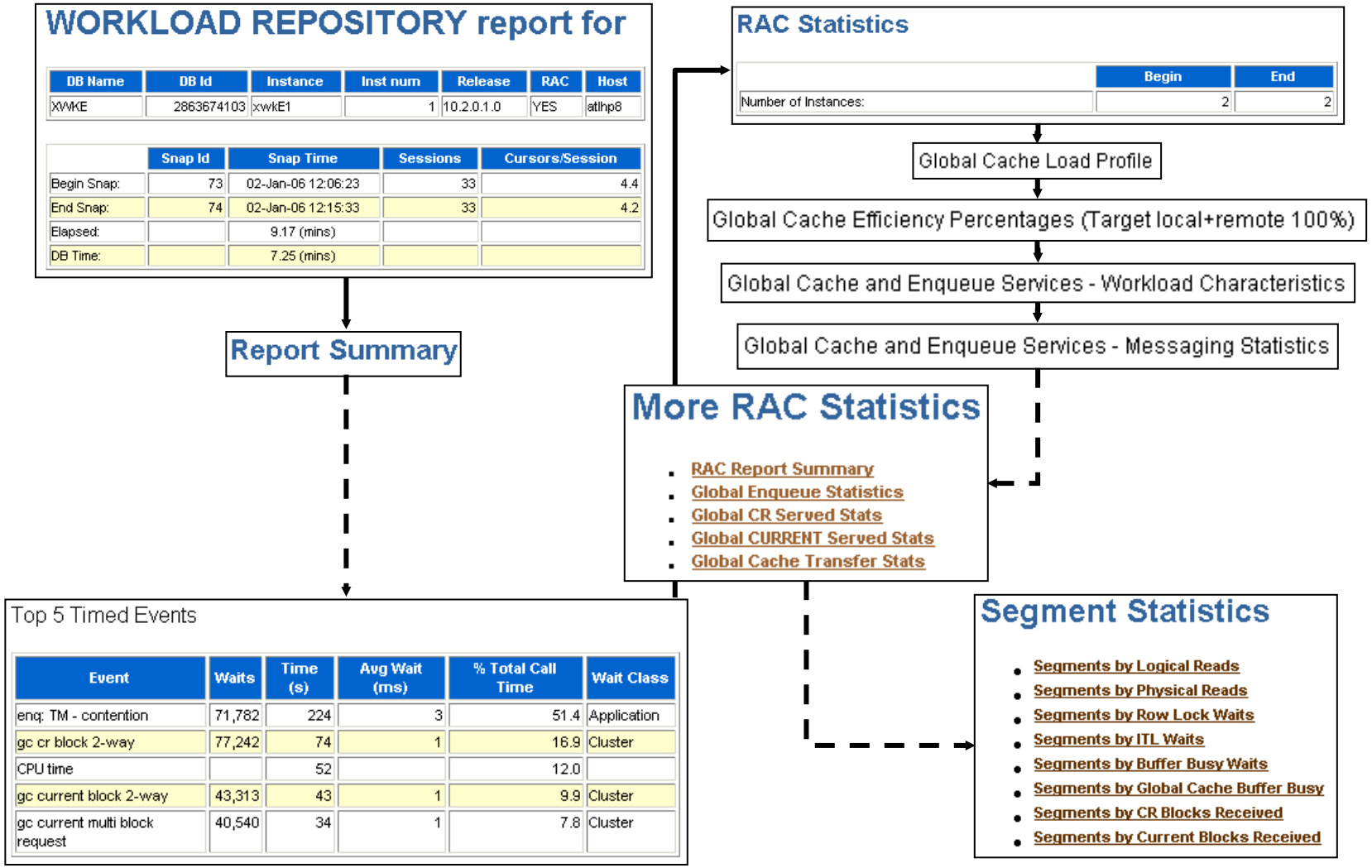
```
select name,wait_class from v$event_name order by 2;
```

AWR RAC 特定环节

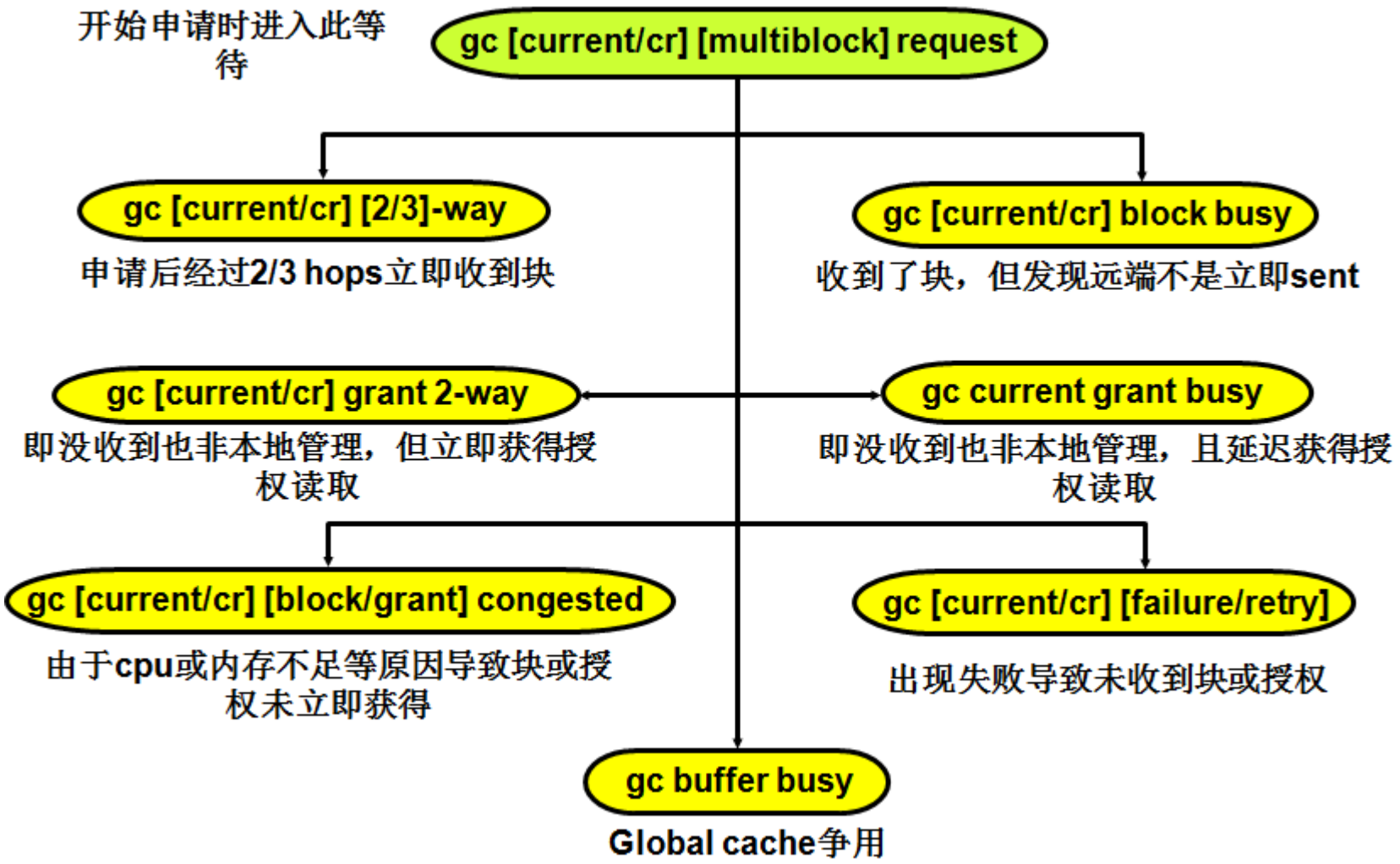
RAC cache fusion

我为人人，
人人为我

预备知识-纵览RAC AWR



预备知识-Global Cache等待事件



Global Cache Transfer就像巧克力，若不打开，你永远不知道下一颗是什么？

RAC Global Cache Load Profile（RAC特有)

RAC Statistics

	Begin	End
Number of Instances:	2	2

Global Cache Load Profile

	Per Second	Per Transaction
Global Cache blocks received:	454.04	1.97
Global Cache blocks served:	447.57	1.94
GCS/GES messages received:	1,123.87	4.87
GCS/GES messages sent:	1,596.69	6.92
DBWR Fusion writes:	31.04	0.13
Estd Interconnect traffic (KB)	7,744.30	

Global Cache Efficiency Percentages (Target local+remote 100%)

Buffer access - local cache %:	99.74
Buffer access - remote cache %:	0.03
Buffer access - disk %:	0.23

真实大型SIEBEL应用的一个例子，0.03%的的逻辑读由其他节点上的缓存满足
Global Cache Receive/server的原因一般是节点间缓存争用或本地无此缓存

(received+send)*db_block_size= 901 * 8k= 7.04MB/s= 56.32Mb/s
注意这仅仅是粗略估算，一般建议private network选用10Gb带宽

一条GCS/GES Message大约200 bytes

RAC Global Cache Load Profile（RAC特有)

Global Cache Load Profile

	Per Second	Per Transaction
Global Cache blocks received:	454.04	1.97
Global Cache blocks served:	447.57	1.94
GCS/GES messages received:	1,123.87	4.87
GCS/GES messages sent:	1,596.69	6.92
DBWR Fusion writes:	31.04	0.13
Estd Interconnect traffic (KB)	7,744.30	

评估的发送流量(MB) = (((GES_SENT+GCS_SENT)*200) + ((CR_SENT+CURRENT_SENT)* BLOCK_SIZE)) * 1.2 * 8 /1048576

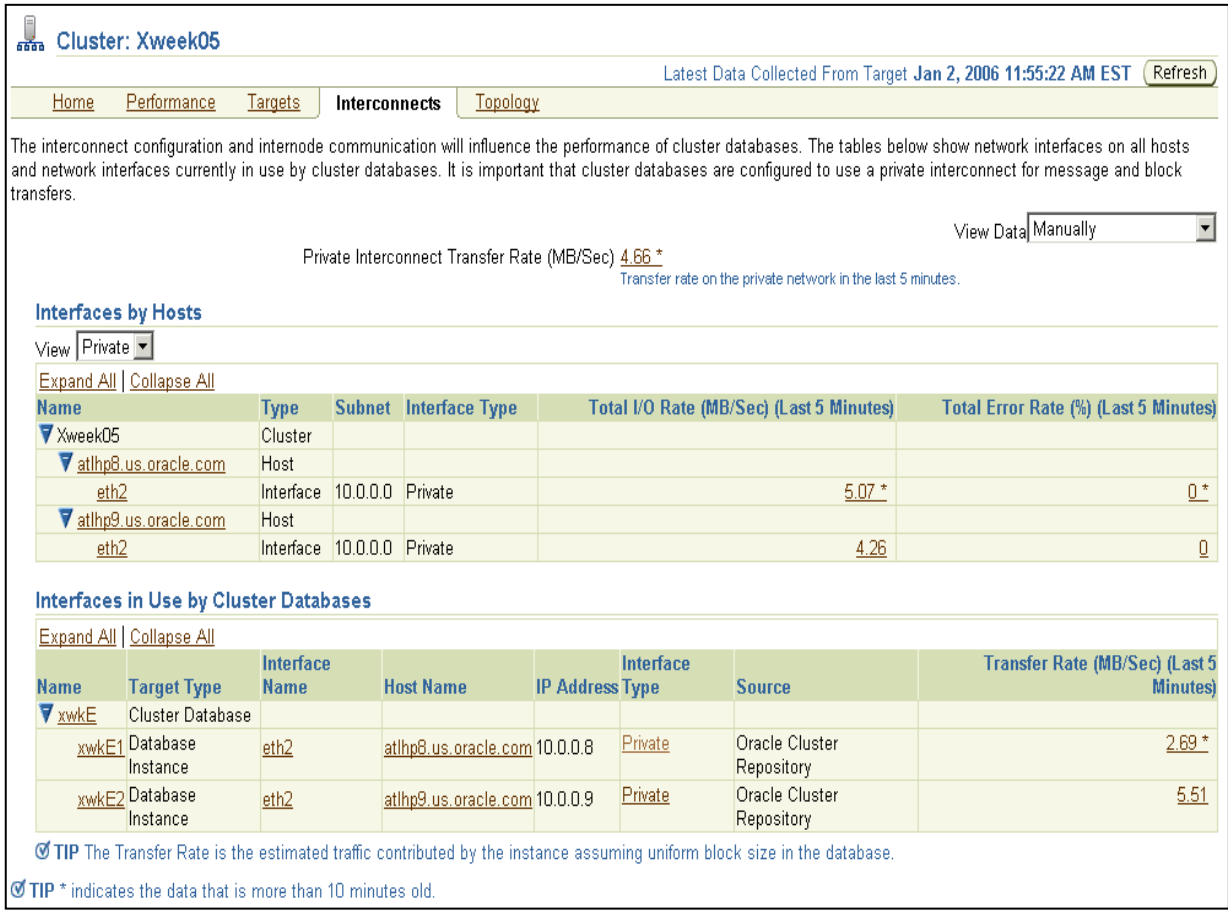
Estd Interconnect traffic (KB) = (('gc cr blocks received'+ 'gc current blocks received' + 'gc cr blocks served'+ 'gc current blocks served') * Block size)
+ (('gcs messages sent' + 'ges messages sent' + 'gcs msgs received'+ 'gcs msgs received')*200)/1024/Elapsed Time

11g DBA_HIST_IC_DEVICE_STATS、 DBA_HIST_CLUSTER_INTERCON

反应Private Network性能的2个维度：
a)可用带宽
b)网络延迟

EM 监控Cluster Interconnect

- 比你手动去查效率高
- 并可定制interconnect 告警



Cluster: Xweek05

Latest Data Collected From Target **Jan 2, 2006 11:55:22 AM EST** Refresh

Home | Performance | Targets | **Interconnects** | Topology

The interconnect configuration and internode communication will influence the performance of cluster databases. The tables below show network interfaces on all hosts and network interfaces currently in use by cluster databases. It is important that cluster databases are configured to use a private interconnect for message and block transfers.

View Data: Manually

Private Interconnect Transfer Rate (MB/Sec) 4.66 *
Transfer rate on the private network in the last 5 minutes.

Interfaces by Hosts

View: Private

Expand All | Collapse All

Name	Type	Subnet	Interface Type	Total I/O Rate (MB/Sec) (Last 5 Minutes)	Total Error Rate (%) (Last 5 Minutes)
Xweek05	Cluster				
atlhp8.us.oracle.com	Host				
eth2	Interface	10.0.0.0	Private	5.07 *	0 *
atlhp9.us.oracle.com	Host				
eth2	Interface	10.0.0.0	Private	4.26	0

Interfaces in Use by Cluster Databases

Expand All | Collapse All

Name	Target Type	Interface Name	Host Name	IP Address	Interface Type	Source	Transfer Rate (MB/Sec) (Last 5 Minutes)
xwkE	Cluster Database						
xwkE1	Database Instance	eth2	atlhp8.us.oracle.com	10.0.0.8	Private	Oracle Cluster Repository	2.69 *
xwkE2	Database Instance	eth2	atlhp9.us.oracle.com	10.0.0.9	Private	Oracle Cluster Repository	5.51

TIP The Transfer Rate is the estimated traffic contributed by the instance assuming uniform block size in the database.

TIP * indicates the data that is more than 10 minutes old.

Global Cache Efficiency Percentages (Target local+remote 100%) （RAC特有)

Global Cache Efficiency Percentages (Target local+remote 100%)

Buffer access - local cache %:	99.74
Buffer access - remote cache %:	0.03
Buffer access - disk %:	0.23

Global Cache and Enqueue Services - Workload Characteristics

Avg global enqueue get time (ms):	0.3
Avg global cache cr block receive time (ms):	0.7
Avg global cache current block receive time (ms):	1.0
Avg global cache cr block build time (ms):	0.0
Avg global cache cr block send time (ms):	0.0
Global cache log flushes for cr blocks served %:	6.8
Avg global cache cr block flush time (ms):	2.9
Avg global cache current block pin time (ms):	0.0
Avg global cache current block send time (ms):	0.0
Global cache log flushes for current blocks served %:	12.4
Avg global cache current block flush time (ms):	3.2

对RAC而言response time响应时间的要求比单节点更高，所以别指望用烂硬件搭出来的RAC性能比单机好 ☺！

如果Interconnect的网络延迟 > IO子系统的延迟，那么RAC本身就是性能瓶颈

但是IO响应时间对RAC也非常重要，例如上一讲中所述log file sync=> gc buffer busy，所以千万别用garbage storage去搭RAC!!

Global Cache Efficiency Percentages (Target local+remote 100%) （RAC特有)

Global Cache Efficiency Percentages (Target local+remote 100%)

Buffer access - local cache %:	99.74
Buffer access - remote cache %:	0.03
Buffer access - disk %:	0.23

Global Cache and Enqueue Services - Workload Characteristics

Avg global enqueue get time (ms):	0.3
Avg global cache cr block receive time (ms):	0.7
Avg global cache current block receive time (ms):	1.0
Avg global cache cr block build time (ms):	0.0
Avg global cache cr block send time (ms):	0.0
Global cache log flushes for cr blocks served %:	6.8
Avg global cache cr block flush time (ms):	2.9
Avg global cache current block pin time (ms):	0.0
Avg global cache current block send time (ms):	0.0
Global cache log flushes for current blocks served %:	12.4
Avg global cache current block flush time (ms):	3.2

Avg global cache cr block receive time (ms):0.7
Avg global cache current block receive time (ms):1.0

至关重要的2个指标，结合其他节点的AWR报告一起分析这2个指标， 一般要求小于2ms

若在RAC实例之间这2个指标差异很大，一般说明interconnect问题出现于OS buffer层或者网卡上

Avg global cache cr block receive time (ms)

该指标反映平均每个global cr块从申请到收到的耗时

Global Cache and Enqueue Services - Workload Characteristics

Avg global enqueue get time (ms):	1.6
Avg global cache cr block receive time (ms):	16.9
Avg global cache current block receive time (ms):	1.5
Avg global cache cr block build time (ms):	0.0
Avg global cache cr block send time (ms):	0.0
Global cache log flushes for cr blocks served %:	0.0
Avg global cache cr block flush time (ms):	0.6
Avg global cache current block pin time (ms):	0.0
Avg global cache current block send time (ms):	0.0
Global cache log flushes for current blocks served %:	0.0
Avg global cache current block flush time (ms):	0.7

gc blocks compressed	188,353	35.84	81.22
gc cr block flush time	3	0.00	0.00
gc cr block receive time	228,252	43.43	98.43
gc cr blocks received	134,978	25.68	58.21
gc cr blocks served	248,969	47.37	107.36
gc current block flush time	3	0.00	0.00
gc current block pin time	2,294	0.44	0.99
gc current block receive time	116,862	22.24	50.39

Avg global cache cr block receive time (ms)= 10 * gc cr block receive time / gc cr blocks received = 228,252 / 134,978 * 10 = 16.91ms

Time to process CR block request in the cache = (build time + flush time + send time)

相关指标：

- gc cr block flush time
- gc cr block build time
- gc cr block send time

Avg global cache current block receive time (ms)

该指标反映平均每个global current块从申请到收到的耗时

Global Cache and Enqueue Services - Workload Characteristics

Avg global enqueue get time (ms):	1.6
Avg global cache cr block receive time (ms):	16.9
Avg global cache current block receive time (ms):	1.5
Avg global cache cr block build time (ms):	0.0
Avg global cache cr block send time (ms):	0.0
Global cache log flushes for cr blocks served %:	0.0
Avg global cache cr block flush time (ms):	0.6
Avg global cache current block pin time (ms):	0.0
Avg global cache current block send time (ms):	0.0
Global cache log flushes for current blocks served %:	0.0

gc cr blocks served	248,969	47.37	107.36
gc current block flush time	3	0.00	0.00
gc current block pin time	2,294	0.44	0.99
gc current block receive time	116,862	22.24	50.39
gc current blocks received	790,899	150.49	341.05
gc current blocks served	999,835	190.25	431.15
gc kbytes saved	508,740	96.80	219.38
gc kbytes sent	9,481,676	1,804.16	4,088.69
gc local grants	8,062,933	1,534.20	3,476.90
gc read wait time	199,673	37.99	86.10
gc read wait timeouts	48	0.01	0.02

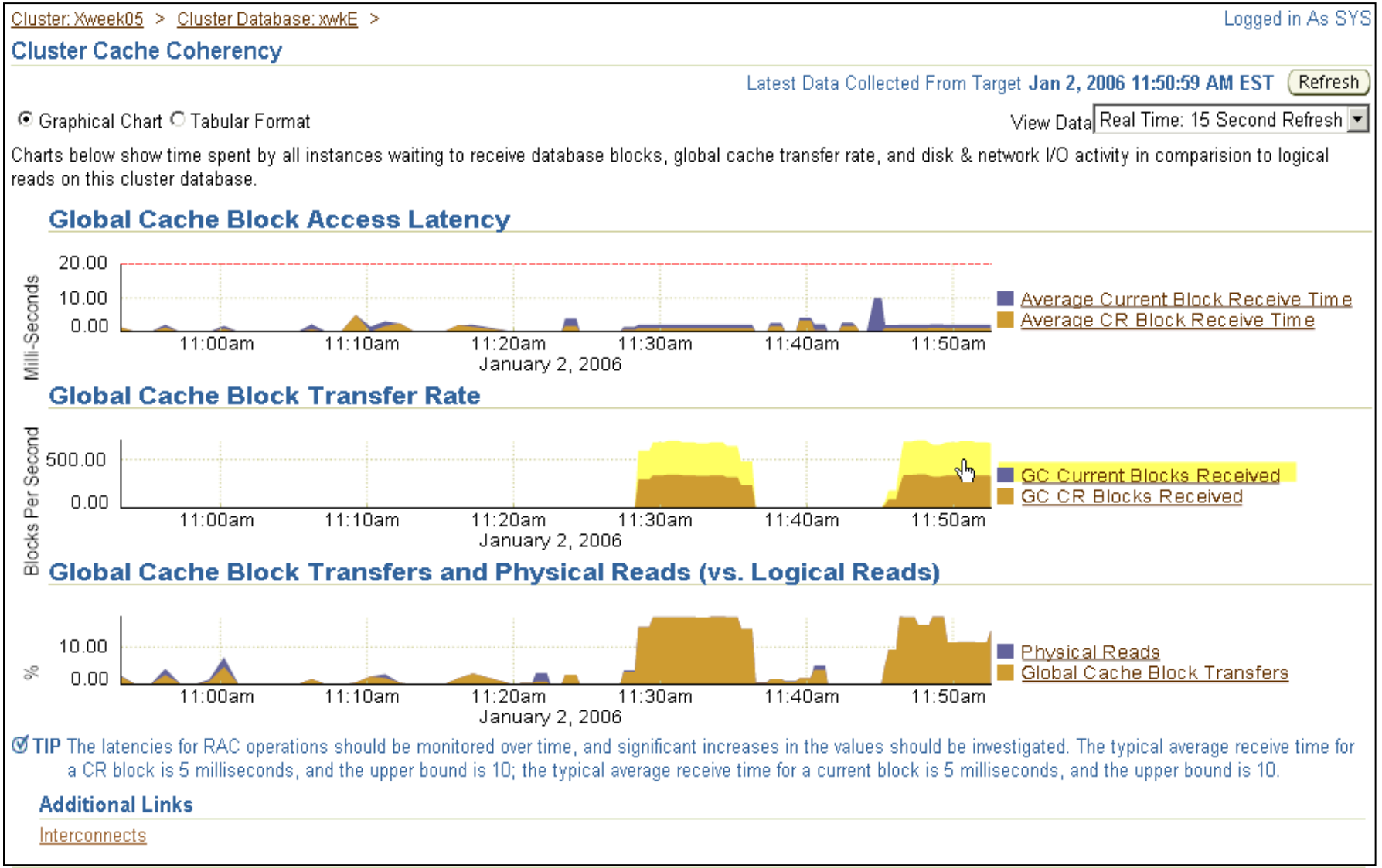
Avg global cache cr block receive time (ms)= 10 * gc current block receive time / gc current blocks received = 116,862 / 790,899 * 10 = 1.47ms

Time to process current block request in the cache= (pin time + flush time + send time)

相关指标：

- gc current block pin time
- gc current block flush time
- gc current block send time

Cluster Cache Coherency



Apex-High Cluster Wait真实案例

Top 5 Timed Events

Event	Waits	Time (s)	Avg wait (ms)	%Total Call Time	Wait Class
gc buffer busy	79,083	73,024	923	65.4	Cluster
enq: TX - row lock contention	35,068	17,123	488	15.3	Applicatio
CPU time		12,205		10.9	
gc current request	2,714	3,315	1221	3.0	Cluster
gc cr multi block request	83,666	1,008	12	0.9	Cluster

Global Cache and Enqueue Services - Workload Characteristics

Avg global enqueue get time (ms):	1.7
Avg global cache cr block receive time (ms):	4.5
Avg global cache current block receive time (ms):	4.7
Avg global cache cr block build time (ms):	0.0
Avg global cache cr block send time (ms):	0.0
Global cache log flushes for cr blocks served %:	1.7
Avg global cache cr block flush time (ms):	20.7
Avg global cache current block pin time (ms):	201.5
Avg global cache current block send time (ms):	0.0
Global cache log flushes for current blocks served %:	0.1
Avg global cache current block flush time (ms):	174.4

过高

影响其他节点性能

Avg global cr/current block receive time高➔ 本实例出现大量cluster等待

Avg global cache pin/send/flush time高➔ 本实例server global cache也不给力啊，其他节点的性能大致也和此处一副德性

Apex-High Cluster Wait真实案例

Global Cache and Enqueue Services - Messaging Statistics

```
        Avg message sent queue time (ms):    205.4
Avg message sent queue time on ksxp (ms):     24.8
    Avg message received queue time (ms):      0.0
       Avg GCS message process time (ms):      0.0
       Avg GES message process time (ms):      0.0

          % of direct sent messages:    51.56
        % of indirect sent messages:    37.38
       % of flow controlled messages:    11.07
```

Avg message sent queue time　　一条信息进入队列到发送它的时间
Avg message sent queue time on ksxp　对端收到该信息并返回ACK的时间，这个指标很重要，直接反应了网络延迟，一般小于1ms

Send message有2种方式：
kjccsmg() – send message (FG direct send)
kjccqmg() – queue message (indirect send by LMS)

% of indirect sent messages ➔ 间接发送信息一般是排序或大的信息，流控制也可能引起indirect sent message
% of flow controlled messages ➔ 流控制最常见的原因是网络状况不佳， % of flow controlled messages应当小于1%

Apex-High Cluster Wait真实案例

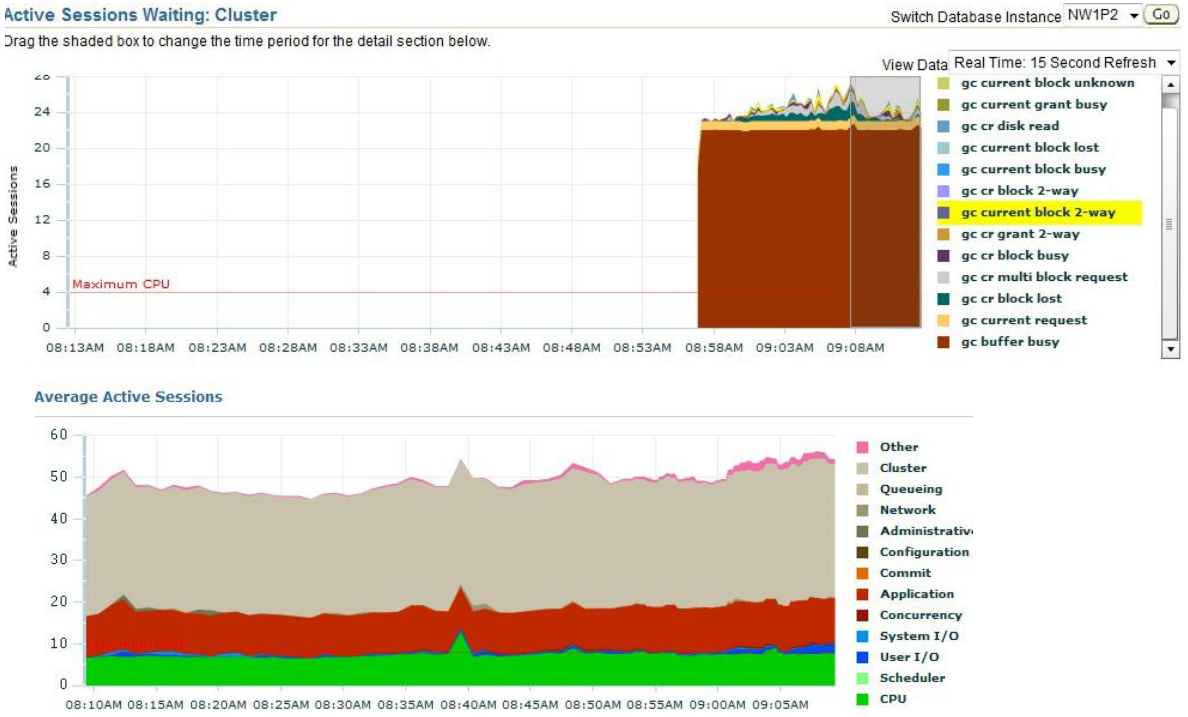
Wait Class	Waits	%Time -outs	Total Wait Time (s)	Avg wait (ms)	Waits /txn
Cluster	324,359	23.8	78,259	241	9.2
Application	37,448	93.2	17,128	457	1.1
Other	1,739,365	95.7	815	0	49.4
User I/O	219,779	.0	740	3	6.2
Concurrency	49,438	.3	262	5	1.4
System I/O	52,399	.0	211	4	1.5
Network	775,285	.0	137	0	22.0
Commit	6,635	.4	128	19	0.2
Configuration	3,981	1.1	82	21	0.1
Administrative	1	100.0	0	12	0.0

Event	Waits	%Time -outs	Total Wait Time (s)	Avg wait (ms)	Waits /txn
gc buffer busy	79,083	94.2	73,024	923	2.2
enq: TX - row lock contentio	35,068	99.5	17,123	488	1.0
gc current request	2,714	99.6	3,315	1221	0.1
gc cr multi block request	83,666	.1	1,008	12	2.4
db file sequential read	115,086	.0	593	5	3.3
cr request retry	1,449	100.0	233	161	0.0
gc cr block lost	293	1.7	200	684	0.0
gc cr block busy	18,367	.1	196	11	0.5
PX Deq: reap credit	71,527	82.9	175	2	2.0
DBMS_LDAP: LDAP operation	2,988	.0	149	50	0.1
log file parallel write	27,067	.0	138	5	0.8
log file sync	6,635	.4	128	19	0.2
gc cr block 2-way	48,805	.0	119	2	1.4

大量Cluster Wait，平均每次等241ms，而一个事务平均等9.2次，则一个事务平均等Cluster 2秒，换句话说如果不是RAC，每个事务提速2s!!

Cluster Wait并不孤立，会影响Application类型的enq: TX - row lock contention等待事件出现的频率和单次等待耗时

如上例中 平均 每个事务要等一次的enq: TX - row lock contentio，每次平均耗时0.5s，100~200个session等在enq: TX 和 gc buffer busy上很壮观哦！



gcs resource directory to be unfrozen真实案例

Top 5 Timed Foreground Events

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
gcs resource directory to be unfrozen	37,184	71,204	1915	44.49	Other
library cache pin	1,609	67,408	41894	42.12	Concurrency
buffer busy waits	87,752	8,638	98	5.40	Concurrency
ges enter server mode	7,340	3,406	464	2.13	Other
log file sync	102,808	2,253	22	1.41	Commit

Dynamic Remastering Stats

- times are in seconds
- Affinity objects - objects mastered due to affinity at begin/end snap

Name	Total	per Remaster Op	Begin Snap	End Snap
remaster ops	1	1.00		
remastered objects	1	1.00		
replayed locks received	38,019,474	38,019,474.00		
replayed locks sent	17	17.00		
resources cleaned	0	0.00		
remaster time (s)	92.7	92.66		
quiesce time (s)	18.6	18.63		
freeze time (s)	0.0	0.01		
cleanup time (s)	7.1	7.06		
replay time (s)	0.1	0.07		
fixwrite time (s)	66.8	66.82		
sync time (s)	0.0	0.00		
affinity objects			611	919

Global Cache and Enqueue Services - Messaging Statistics

Avg message sent queue time (ms):	915.6
Avg message sent queue time on ksxp (ms):	3.8
Avg message received queue time (ms):	77.8
Avg GCS message process time (ms):	0.0
Avg GES message process time (ms):	1.3
% of direct sent messages:	0.09
% of indirect sent messages:	99.91
% of flow controlled messages:	0.00

gcs resource directory to be unfrozen与DRM 有直接关系

某银行 gc cr block lost真实案例

Top 5 Timed Events

Event	Waits	Time(s)	Avg Wait(ms)	% Total Call Time	Wait Class
gc cr block lost	24,610	26,896	1,093	40.2	Cluster
db file sequential read	29,614,961	19,228	1	28.8	User I/O
gc buffer busy	1,722,143	8,887	5	13.3	Cluster
read by other session	16,660,232	7,999	0	12.0	User I/O
CPU time		4,455		6.7	

诊断RAC全局缓存块丢失gc blocks lost

2011/01/05 BY MACLEAN LIU 1 COMMENT

在Oracle RAC环境中，无论我们从AWR自动负载性能报告、Statspack或者Grid Control中都可以找到Oracle数据库软件所收集的全局缓存工作负载统计信息(global cache work load statistics)；其中就包含了全局缓存块丢失(Global cache lost blocks)的统计信息(这些丢失的全局缓存块可能是gc cr block lost或者gc current block lost)，若集群中的任意节点出现大量的全局缓存块丢失(下文简写为gc blocks lost)，则可能意味着内联(private)网络存在问题或者packet网络包处理低效。通过监控和评估这些全局缓存的相关统计信息，可以有效保证内联全局缓存(interconnect Global Cache)和全局队列服务(Global Enqueue Service (GCS/GES)以及整个集群的正常工作。全局缓存块丢失一般预示着网络包处理存在问题并需要进一步勘察。另外全局缓存块丢失(gc blocks lost)的问题常会伴随着gc cr multiblock waits等待发生(传输多个连续的数据块全局缓存)。

就目前来看最有嫌疑造成或加速gc blocks lost的"元凶"往往是因为错误地或者不当的配置了内联网络(interconnects)。接下来我们会进一步介绍如何找出造成gc blocks lost的原因。

虽然gc blocks lost对集群造成的影响更多的反应在性能方面，但我们也无法保证其没有造成节点/实例被驱逐(eviction)的可能性。Oracle Clusterware集群及Oracle RAC实例的节点成员管理依赖于内联网络的心跳(heartbeats)。假设在网络心跳持续丢失的情况下，节点/实例的驱逐可以发生。以下我们列出gc blocks lost可能造成的主次要症状:

Global Cache and Enqueue Services - Messaging Statistics

Avg message sent queue time (ms):	0.1
Avg message sent queue time on ksxp (ms):	0.4
Avg message received queue time (ms):	19.6
Avg GCS message process time (ms):	0.0
Avg GES message process time (ms):	0.0
% of direct sent messages:	58.11
% of indirect sent messages:	41.23
% of flow controlled messages:	0.66

某银行 gc cr block lost真实案例

Top 5 Timed Events

Event	Waits	Time(s)	Avg Wait(ms)	% Total Call Time	Wait Class
gc cr block lost	24,610	26,896	1,093	40.2	Cluster
db file sequential read	29,614,961	19,228	1	28.8	User I/O
gc buffer busy	1,722,143	8,887	5	13.3	Cluster
read by other session	16,660,232	7,999	0	12.0	User I/O
CPU time		4,455		6.7	

```
rac1: $netstat -in
Name      Mtu  Network       Address           Ipkts    Ierrs Opkts    Oerrs Coll
lan2*     1500 none          none              0        0     0        0     0
lan1:1    1500 160.0.0.0     160.0.0.143       28056153 0      940798   0      0
lan1      1500 160.0.0.0     160.0.0.141       18727524 0      101346403 0      0
lan0      1500 192.168.50.0  192.168.50.10     393487   0      220210   0      0
lo0       4136 127.0.0.0     127.0.0.1         15463444 0      15463629 0      0
lan900    1500 172.168.50.0  172.168.50.10     44797646 9361   123304563 0      0
```

Ierrs is the number of received packets that the system recognized as being corrupted.

Global Cache问题一定要和OS和Network层结合起来看，ifconfig、netstat、syslog信息极易获得，nmon、osw亦有价值

某果的gc buffer busy acquire案例

Top 5 Timed Foreground Events

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
DB CPU		10,135		36.38	
gc buffer busy acquire	322,783	4,327	13	15.53	Cluster
direct path write	48,118	2,719	57	9.76	User I/O
cell single block physical read	1,316,965	2,362	2	8.48	User I/O
gc current block busy	195,130	2,347	12	8.42	Cluster

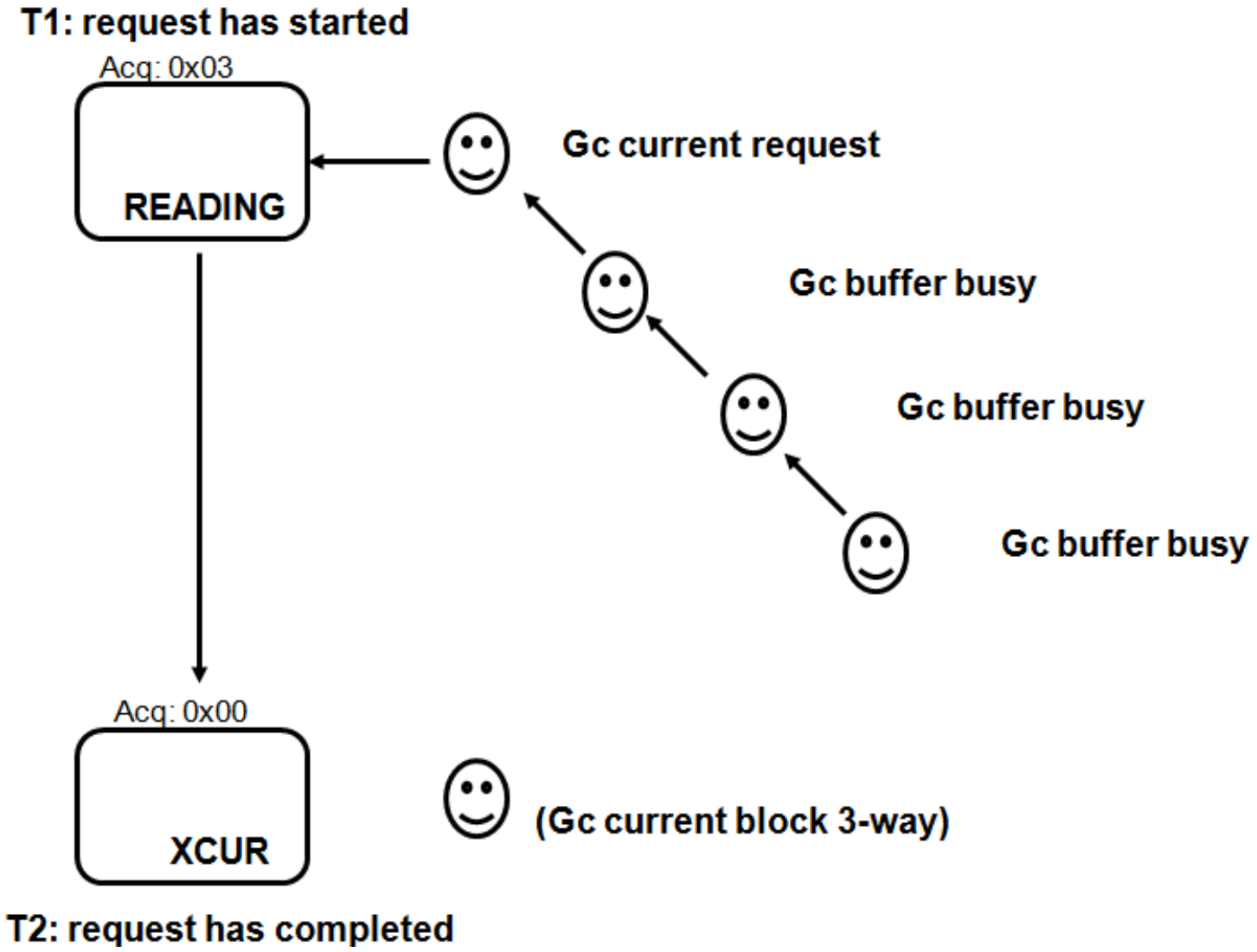
SQL ordered by Cluster Wait Time

- %Total - Cluster Time as a percentage of Total Cluster Wait Time
- %Clu - Cluster Time as a percentage of Elapsed Time
- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Only SQL with Cluster Wait Time > .005 seconds is reported
- Total Cluster Wait Time (s): 9,872
- Captured SQL account for 94.9% of Total

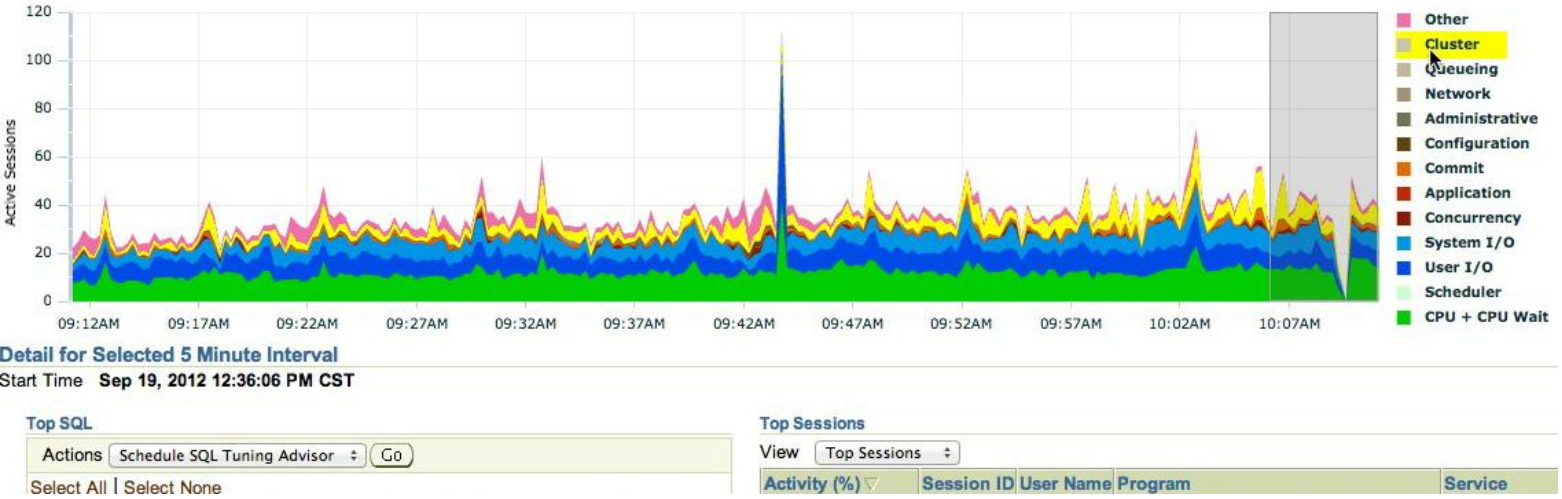
Cluster Wait Time (s)	Executions	%Total	Elapsed Time(s)	%Clu	%CPU	%IO	SQL Id	SQL Module	SQL
5,194.65	130,916	52.62	6,056.07	85.78	1.50	0.73	dhp00aar8kuun	JDBC Thin Client	INSERT INTO Produc
1,237.46	127,236	12.53	2,924.59	42.31	46.81	3.71	4x7qzdxsd0hnn	JDBC Thin Client	begin pkg_parametri
1,000.30	127,234	10.13	1,128.42	88.65	5.74	0.91	52tdjf8ht1tx2	JDBC Thin Client	UPDATE INDIGO.PAI

RAC负载均衡下并发insert很容易造成gc buffer busy acquire
本实例第一个申请该current block时进入gc current request，本实例后续其他进程若又发起对该current block的申请则均进入’gc buffer busy acquire’等待。
11g后gc buffer busy分成 ’gc buffer busy acquire’ 和 ’gc buffer busy release了’

某果的gc buffer busy acquire案例



某果的gc buffer busy acquire案例



人家监控用机的OS X Mountain Lion 哦

Bug 12595929 : GC BUFFER BUSY ACQUIRE CAUSING TOO SLOW PERFORMANCE DURING INSERTS IN 11.2.0.2.2

某大型视频网站gc buffer busy release/acquire案例

Top 5 Timed Foreground Events

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
enq: TX - row lock contention	170,732	76,510	448	41.21	Application
gc buffer busy release	311,756	26,611	85	14.33	Cluster
gc buffer busy acquire	598,333	20,626	34	11.11	Cluster
cell single block physical read	3,588,957	14,903	4	8.03	User I/O
log file sync	342,534	13,630	40	7.34	Commit

Global Cache and Enqueue Services - Workload Characteristics

Avg global enqueue get time (ms):	41.0
Avg global cache cr block receive time (ms):	6.0
Avg global cache current block receive time (ms):	8.9
Avg global cache cr block build time (ms):	0.0
Avg global cache cr block send time (ms):	0.0
Global cache log flushes for cr blocks served %:	9.7
Avg global cache cr block flush time (ms):	34.8
Avg global cache current block pin time (ms):	8.4
Avg global cache current block send time (ms):	0.0
Global cache log flushes for current blocks served %:	9.9
Avg global cache current block flush time (ms):	37.4

Redo flush慢造成的**gc buffer busy release/acquire**

某大型视频网站gc buffer busy release/acquire案例

Global CURRENT Served Stats

- Pins = CURRENT Block Pin Operations
- Flushes = Redo Flush before CURRENT Block Served Operations
- Writes = CURRENT Block Fusion Write Operations

Statistic	Total	% <1ms	% <10ms	% <100ms	% <1s	% <10s
Pins	1,619,064	95.52	3.90	0.55	0.04	0.00
Flushes	160,614	0.98	19.35	74.81	4.86	0.00
Writes	260,966	0.00	10.44	86.53	3.04	0.00

Global Cache Transfer Stats

- Immediate (Immed) - Block Transfer NOT impacted by Remote Processing Delays
- Busy (Busy) - Block Transfer impacted by Remote Contention
- Congested (Congst) - Block Transfer impacted by Remote System Load
- ordered by CR + Current Blocks Received desc

		CR				Current			
Inst No	Block Class	Blocks Received	% Immed	% Busy	% Congst	Blocks Received	% Immed	% Busy	% Congst
3	data block	424,231	87.03	12.85	0.12	914,124	86.86	12.99	0.13
4	data block	229,574	88.98	10.92	0.10	608,464	87.44	12.44	0.12
3	undo header	76,153	88.58	11.33	0.09	1,205	81.83	18.09	0.08
3	Others	3,333	78.94	20.91	0.15	50,605	46.99	52.95	0.06
4	undo header	44,619	92.11	7.79	0.10	535	91.03	8.97	0.00
4	Others	1,365	84.76	15.09	0.15	22,543	66.34	33.58	0.08
3	undo block	13,362	71.92	28.03	0.05	0			
1	data block	2,675	91.55	8.45	0.00	5,719	84.56	15.33	0.09
4	undo block	7,218	74.27	25.66	0.07	0			
1	undo header	1,196	97.32	2.51	0.17	4	100.00	0.00	0.00
1	Others	51	98.04	1.96	0.00	458	43.89	56.11	0.00
1	undo block	99	68.69	31.31	0.00	0			

Global Cache Transfer Times (ms)

- Avg Time - average time of all blocks (Immed,Busy,Congst) in ms
- Immed, Busy, Congst - Average times in ms
- ordered by CR + Current Blocks Received desc

		CR Avg Time (ms)				Current Avg Time (ms)			
Inst No	Block Class	All	Immed	Busy	Congst	All	Immed	Busy	Congst
3	data block	7.22	0.28	54.31	0.34	8.20	0.46	60.04	0.43
4	data block	4.16	0.26	35.97	0.29	7.84	0.44	59.95	0.46
3	undo header	5.34	0.22	45.43	0.26	8.66	0.41	45.99	0.18
3	others	12.40	0.32	58.11	0.39	30.52	8.19	50.36	3.27
4	undo header	2.82	0.20	33.78	0.25	3.18	0.28	32.66	
4	others	6.51	0.25	41.72	0.50	14.08	1.16	39.63	1.88
3	undo block	13.25	0.23	46.68	0.30				
1	data block	2.09	0.26	21.92		11.27	0.30	71.80	0.26
4	undo block	8.59	0.2	32.87	0.27				
1	undo header	0.59	0.17	16.84	0.16	0.20	0.20		
1	others	0.42	0.33	5.11		23.53	0.83	41.29	
1	undo block	6.45	0.13	20.19					

由于gc buffer busy 35~50ms才能收到一个block，这延迟。。。
打 dota是够了，当然前提是你APM够低哦。。。

AWR SQL Statistics

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
- Total DB Time (s): 464
- Captured SQL account for 194.8% of Total

Elapsed Time (s)	CPU Time (s)	Executions	Elap per Exec (s)	% Total DB Time	SQL Id	SQL Module	SQL Text
448	459	1	448.46	96.58	a62jx4x96cuq7	sqlplus@vrh8.oracle.com (TNS V1-V3)	declare v_cnt int; begin for i...
448	459	500	0.90	96.56	2q7cjfag1c50r	sqlplus@vrh8.oracle.com (TNS V1-V3)	select count(*) from mac_large
3	3	1	3.15	0.68	572fbaj0fdw2b	sqlplus@vrh8.oracle.com (TNS V1-V3)	select output from table(dbms_...
2	2	2	0.98	0.42	1uk5m5qbzj1vt	sqlplus@vrh8.oracle.com (TNS V1-V3)	BEGIN dbms_workload_repository...
0	0	2	0.24	0.10	d92h3rjp0y217		begin prvt_hdm.auto_execute(:...
0	0	2	0.16	0.07	bunssq950snhf		insert into wrh$_sga_target_ad...
0	0	2	0.12	0.05	350myuyx0t1d6		insert into wrh$_tablespace_st...
0	0	2	0.07	0.03	fktqvw2wjxdxc		insert into wrh$_filestatxs (s...
0	0	2	0.05	0.02	35d1p0whzav6x		insert into wrh$_tempstatxs (s...
0	0	839	0.00	0.02	96g93hntrzjtr		select /*+ rule */ bucket_cnt,...

注意对于PL/SQL，SQL Statistics不仅会体现该PL/SQL的执行情况，还会包括该PL/SQL包含的SQL语句的情况。如上例一个TOP PL/SQL执行了448s，而这448s中绝大多数是这个PL/SQL下的一个SQL执行500次耗费的。

则该TOP PL/SQL和TOP SQL都上榜，一个执行一次耗时448s，一个执行500次耗时448s。 如此情况则Elapsed Time加起来可能超过100%的Elapsed Time，这是正常的。

AWR SQL ordered by Elapsed Time真实案例

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
- %Total - Elapsed Time as a percentage of Total DB time
- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Captured SQL account for 94.8% of Total DB Time (s): 3,663,161
- Captured PL/SQL account for 1.1% of Total DB Time (s): 3,663,161

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module	SQL Text
3,434,416.87	0		93.76	48.10	0.28	10vahrk09d68a	1459965.1	INSERT /*+ PARALLEL(afp DEFAUL...
14,325.12	1	14,325.12	0.39	26.50	32.21	229z8manccf3d	1459961.1	begin :retval := pkg_ln_dly_ac...
14,321.32	1	14,321.32	0.39	26.48	32.22	4s98dy3jw3u7v	1459961.1	INSERT /*+ PARALLEL(psp DEFAUL...
12,427.76	1	12,427.76	0.34	20.41	25.16	3wbdbjnbukgdg	1459961.1	begin :retval := pkg_ln_dly_ac...
12,226.57	1	12,226.57	0.33	20.45	24.30	bnspg14f5p8m5	1459961.1	INSERT /*+ PARALLEL(pslr DEFAU...
6,398.48	0		0.17	60.16	3.80	b2q0h8z03hs6x	1459961.1	begin :retval := pkg_ln_dly_ac...
6,393.21	0		0.17	60.18	3.78	88zrv7wydc0z1	1459961.1	INSERT /*+ PARALLEL(pslcr DEFA...
3,762.17	1	3,762.17	0.10	25.81	35.02	82m7q4fp51642	1459961.1	begin :retval := pkg_ln_dly_ac...
2,979.92	12	248.33	0.08	24.23	35.12	12xztf70khh8m	1459961.1	INSERT /*+ PARALLEL(pshl DEFAU...
984.81	1	984.81	0.03	15.93	40.04	20gdh69jau473	1459963.1	begin :retval := pkg_acp_cmn_e...

对于鹤立鸡群的SQL很有必要一探究竟，跑个@?/rdbms/admin/awrsqrpt看看吧！

1. %Total - Elapsed Time as a percentage of Total DB time➔这个语句耗总的DB TIME的百分之多少
2. %CPU - CPU Time as a percentage of Elapsed Time ➔ 这个语句耗费的DB TIME里CPU占多少比例➔这个语句是否是CPU敏感的
3. %IO - User I/O Time as a percentage of Elapsed Time ➔ 这个语句耗费的DB TIME里IO占多少比例➔这个语句是否是IO敏感的
4. Captured SQL account for 94.8% of Total DB Time (s) ➔对于不绑定变量的应用来说Top SQL有可能失准，所以要参考本项

超高Version Count真实案例

SQL ordered by Version Count

- Only Statements with Version Count greater than 20 are displayed

Version Count	Executions	SQL Id	SQL Module
8,875		dtxfqv0tujgqd	
6,401		0kxrcgm563dkc	
6,317		bgtwnk9927s3x	
6,000		9xd3gb79z6nn4	
3,980		24sq9764ydzqa	
3,836		abumgdhqpcz0u	
2,543		1d4pb0f64wvp2	
2,244		025wwk233shf4	
2,244		82s6mcbkvtjaf	
2,244		ggvvynud20068	
2,212		98q9xfxz9w0ps	
2,181		dmq5n9ta7xz9t	
1,849		bxr8u17mjj4ub	
1,746		cwx59q6598k3v	
1,694		52rb712rtvgfs	
1,663		8pkdwd4qg0j3z	
1,666		1tcgtu3v82w9d	

Top 5 Timed Foreground Events

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
resmgr:cpu quantum	21,358,967	414,492	19	42.51	Scheduler
cursor: mutex S	93,237,890	311,910	3	31.99	Concurrency
library cache lock	431,205	103,661	240	10.63	Concurrency
cursor: mutex X	53,422,273	48,464	1	4.97	Concurrency
DB CPU		34,239		3.51	

Version Count高易造成一系列cursor sharing性能问题，包括可能的ORA-4031、KKSFBC CHILD COMPLETION、Latch:shared Pool、cursor: mutex S or X

你有必要为超高的version count 提一个Service Request!

RAC 优化感悟 单节点性能优化

一人吃饱全家不饿
硬件有点毛病，多
少都能熬过去，缝
缝补补又三年

单节点对开发和DBA技术水准的要求较低，即便硬件有性能问题往往也能维持工作

RAC 优化感悟 RAC性能优化

爱真的需要勇气和硬件基础，哪一方回馈得慢一点都会让双方尝到苦果，所谓的”贫贱夫妻百事哀”。

RAC对开发和DBA技术水准有一定要求，硬件性能问题往往直接导致HANG、节点重启、响应极缓慢等不幸福现象。

才开了个头哦。。。

To Be Continued……

つづく……

敬请期待开Oracle调优鹰眼，深入理解AWR性能报告第三讲

http://t.askmaclean.com/forum-4-1.html

更多信息

www.askmaclean.com

or

http://www.askmaclean.com/archives/tag/tuning

Question & Answer

If you have more questions later, feel free to ask