【故障处理】队列等待之 enq IV - contention 案例

1.1 BLOG 文档结构图

1.2 前言部分

1.2.1 导读和注意事项

各位技术爱好者，看完本文后，你可以掌握如下的技能，也可以学到一些其它你所不知道的知识，~O(∩_∩)O~：

① 队列等待之 enq IV - contention 案例（重点）

Tips:

① 本文在 itpub（http://blog.itpub.net/26736162）、博客园（http://www.cnblogs.com/lhrbest）和微信公众号（xiaomaimiaolhr）上有同步更新。

② 文章中用到的所有代码、相关软件、相关资料及本文的 pdf 版本都请前往小麦苗的云盘下载，小麦苗的云盘地址见：http://blog.itpub.net/26736162/viewspace-1624453/。

③ 若网页文章代码格式有错乱，请下载 pdf 格式的文档来阅读。

④ 在本篇 BLOG 中，代码输出部分一般放在一行一列的表格中。

本文如有错误或不完善的地方请大家多多指正，ITPUB 留言或 QQ 皆可，您的批评指正是我写作的最大动力。

1.3 故障分析及解决过程

1.3.1 数据库环境介绍

项目	source db
db 类型	RAC
db version	12.1.0.2.0
db 存储	ASM
OS 版本及 kernel 版本	SuSE Linux Enterprise Server (SLES 11) 64 位

1.3.2 AWR 分析

Database Summary

Database				Snapshot Ids		Number of Instances		Number of Hosts		Report Total (minutes)	
Id	Name	RAC	Block Size	Begin	End	In Report	Total	In Report	Total	DB time	Elapsed time
521111702	PLHRDB	YES	8192	3440	3441	2	2	2	2	76.79	59.81

Database Instances Included In Report

- Listed in order of instance number, I#

I#	Instance	Host	Startup	Begin Snap Time	End Snap Time	Release	Elapsed Time(min)	DB time(m
1	plhrdb1	HQaPQQ-PLHR-R01	11-8月 -16 20:51	14-12月-16 20:00	14-12月-16 21:00	12.1.0.2.0	59.8(	63
2	plhrdb2	HQaPQQ-PLHR-R02	12-12月-16 02:57	14-12月-16 20:00	14-12月-16 21:00	12.1.0.2.0	59.8(	13

Report Summary

Top ADDM Findings by Average Active Sessions

Finding Name	Avg active sessions of the task	Percent active sessions of finding	Task Name	Begin Snap Time	End Snap Time
特殊的 "其他" 等待事件	1.28	34.87	ADDM:521111702_3441	14-12月-16 20:00	14-12月-16 21:00
特殊的 "其他" 等待事件	1.28	19.66	ADDM:521111702_3441	14-12月-16 20:00	14-12月-16 21:00
"并行" 等待类	1.28	16.62	ADDM:521111702_3441	14-12月-16 20:00	14-12月-16 21:00
共享池闩锁数	1.28	13.58	ADDM:521111702_3441	14-12	14-12

这里简单分析一下 Up Time(hrs)，其它几个指标都很熟悉了。表中的“Up Time(hrs)”代表自数据库实例启动到本次快照结束这段时间的小时数。例如，该 AWR 中数据库实例 1 的启动时间为“2016-08-11 20:51”，快照结束时间为“2016-12-14 21:00”，故“Up Time(hrs)”约为 125.006 天，转换为小时约为 3000.14 小时，如下所示：

```
SYS@lhrdb> SELECT trunc(UP_TIME_D,3),  trunc(trunc(UP_TIME_D,3)*24,2) UP_TIME_HRS FROM (SELECT
(TO_DATE('2016-12-14 21:00', 'YYYY-MM-DD HH24:MI') - TO_DATE('2016-08-11 20:51', 'YYYY-MM-DD HH24:MI'))
UP_TIME_D FROM DUAL);

TRUNC(UP_TIME_D,3) UP_TIME_HRS
------------------ -----------
         125.006     3000.14
```

可以看到节点 1 的负载较大，而 ADDM 中，特殊类的等待事件较多。接下来查看等待事件的部分：

Top Timed Events

- Instance '*' - cluster wide summary
- '*' Waits, %Timeouts, Wait Time Total(s) : Cluster-wide total for the wait event
- '*' 'Wait Time Avg (ms)' : Cluster-wide average computed as (Wait Time Total / Event Waits) in ms
- '*' Summary 'Avg Wait Time (ms)' : Per-instance 'Wait Time Avg (ms)' used to compute the following statistics
- '*' [Avg/Min/Max/Std Dev] : average/minimum/maximum/standard deviation of per-instance 'Wait Time Avg(ms)'
- '*' Cnt : count of instances with wait times for the event

	Wait		Event		Wait Time			Summary Avg Wait Time (ms)				
I#	Class	Event	Waits	%Timeouts	Total(s)	Avg(ms)	%DB time	Avg	Min	Max	Std Dev	Cnt
*	Other	enq: IV - contention	63,234	1.55	1,726.35	27.30	37.47	29.17	22.26	36.08	9.77	2
	Other	DFS lock handle	40,650	2.02	1,632.32	40.16	35.43	92.56	34.55	150.57	82.04	2
		DB CPU			398.33		8.65					2
	Concurrency	library cache lock	9,904	4.96	378.60	38.23	8.22	29.25	7.55	50.95	30.69	2
	Application	enq: RO - fast object reuse	7,326	0.00	321.87	43.93	6.99	44.87	39.98	49.76	6.92	2
	Other	reliable message	3,021	0.00	279.89	92.65	6.07	58.67	14.79	102.56	62.06	2
	Concurrency	library cache pin	5,920	0.00	245.83	41.53	5.34	31.42	6.66	56.17	35.01	2
	Commit	log file sync	120,858	0.00	192.74	1.59	4.18	1.59	1.59	1.60	0.01	2
	Other	enq: FB - contention	15,996	0.00	174.09	10.88	3.78	10.90	10.03	11.76	1.22	2
	Concurrency	row cache lock	34,753	0.00	141.06	4.06	3.06	3.99	3.81	4.18	0.26	2
1	Other	DFS lock handle	38,685	1.49	1,336.45	34.55	35.32					
	Other	enq: IV - contention	40,170	1.25	894.29	22.26	23.63					
	Concurrency	library cache lock	7,000	5.51	356.66	50.95	9.43					
	Other	reliable message	2,680	0.00	274.85	102.56	7.26					
	Concurrency	library cache pin	4,169	0.00	234.17	56.17	6.19					
		DB CPU			210.85		5.57					
	Application	enq: RO - fast object reuse	4,360	0.00	174.29	39.98	4.61					
	Concurrency	row cache lock	23,584	0.00	98.54	4.18	2.60					
	Other	enq: FB - contention	7,890	0.00	92.80	11.76	2.45					
	Commit	log file sync	57,616	0.00	92.10	1.60	2.43					
2	Other	enq: IV - contention	23,064	2.06	832.06	36.08	101.00					
	Other	DFS lock handle	1,965	12.57	295.87	150.57	35.92					
		DB CPU			187.48		22.76					
	Application	enq: RO - fast object reuse	2,966	0.00	147.58	49.76	17.91					
	Other	ges inquiry response	1,442	0.00	135.74	94.13	16.48					
	Commit	log file sync	63,242	0.00	100.64	1.59	12.22					
	Other	enq: FB - contention	8,106	0.00	81.30	10.03	9.87					
	System I/O	log file parallel write	47,652	0.00	57.56	1.21	6.99					
	Concurrency	row cache lock	11,169	0.00	42.52	3.81	5.16					
	Other	enq: PS - contention	1,148	12.46	34.80	30.32	4.22					

可以看到 enq: IV - contention 和 DFS lock handle 等待较为严重。这里需要说一下 enq: IV - contention 这个名称。在 AWR 中，IV 和-的前后都是 1 个空格，而在数据库中记录的是-之后有 2 个空格，如下：

```
enq: IV -- contention
```

所以，采用搜索的时候需要注意。

Top Events

- Top Events by DB Time
- % Activity is the percentage of DB Time due to the event

Event	Event Class	Session Type	% Activity	Avg Active Sessions
DFS lock handle	Other	FOREGROUND	24.05	0.44
enq: IV - contention	Other	BACKGROUND	13.39	0.25
enq: IV - contention	Other	FOREGROUND	11.72	0.21
CPU + Wait for CPU	CPU	BACKGROUND	10.65	0.20
library cache lock	Concurrency	FOREGROUND	6.09	0.11

Back to Active Session History(ASH) Report
Back to Top

Top Event P1/P2/P3 Values

- Top Events by DB Time and the top P1/P2/P3 values for those events.
- % Event is the percentage of DB Time due to the event
- % Activity is the percentage of DB Time due to the event with the given P1,P2,P3 Values.

Event	% Event	P1, P2, P3 Values	% Activity	Parameter 1	Parameter 2
enq: IV - contention	25.11	"1230372867","1398361667","24"	1.07	type|mode	id1
DFS lock handle	24.51	"1128857605","123","1"	17.35	type|mode	id1
		"1128857605","123","5"	6.39		
library cache lock	6.09	"272666734984","280821609928","511659454038019"	0.15	handle address	lock address
reliable message	4.87	"301251386960","302315898592","301251528640"	0.76	channel context	channel hand
enq: RO - fast object reuse	4.57	"1380909062","65848","1"	1.52	name|mode	2
		"1380909058","65848","1"	1.37		

根据 ASH 中的 p1 参数的值获得：

```
SYS@lhrdb> SELECT CHR(BITAND(1230372867, -16777216) / 16777215) ||
  2         CHR(BITAND(1230372867, 16711680) / 65535) "LOCK",
  3         BITAND(1230372867, 65535) "MODE"
  4    FROM DUAL;

LO       MODE
-- ----------
IV          3
```

1.3.3 enq: IV - contention 解决

```
SELECT *
  FROM V$EVENT_NAME A
 WHERE A.NAME LIKE '%enq: IV -  contention%';
```

EVENT#	EVENT_ID	NAME	PARAMETER1	PARAMETER2	PARAMETER3	WAIT_CLASS_ID	WAIT_CLASS#	WAIT_CLASS	DISPLAY_NAME	CON_ID
1195	3753139397	enq: IV - contention	type|mode	id1	id2	1893977003	0	Other	enq: IV - contention	0

该等待事件为 12c 特有，12c 相比 11g 多了大约 500 多个等待事件。该问题参考 MOS：12c RAC DDL Performance Issue: High "enq: IV - contention" etc if CPU Count is Different (文档 ID 2028503.1)

12c RAC DDL Performance Issue High enq IV - contention etc if CPU Count is Different (文档 ID 2028503.1).mhtml

The fix will be included in future PSU, patch exists for certain platform/version.

The workaround is to set the following parameter to the highest value in the cluster and restart:

```
_ges_server_processes
```

To find out the highest value, run the following grep on each node:

```
ps -ef| grep lmd
```

该等待事件主要是由于 12c RAC 的 2 个节点上的 cpu_count 这个变量不一致导致的。

从 AWR 中可以看出节点 1 的 CPU 为 48，而节点 2 的 CPU 为 96。

OS Statistics By Instance

- Listed in order of instance number, I#
- End values are diplayed only if different from begin values

I#	Num CPUs	CPU Cores	CPU Sckts	Load Begin	Load End	% Busy	% Usr	% Sys	% WIO	% Idl	Busy Time (s)	Idle Time
1	48	48	4	0.92	0.66	1.55	1.14	0.39	0.01	98.45	2,659.65	168,935.
2	96	48	4	0.39	0.56	0.49	0.30	0.17	0.01	99.51	1,671.10	342,104.
Sum											4,330.75	511,040.

从 dba_hist_parameter 中可以看到 CPU_COUNT 这个参数的变化历史：

```
SQL> SHOW PARAMETER CPU

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
cpu_count                            integer     96
parallel_threads_per_cpu             integer     2
resource_manager_cpu_allocation      integer     96


SQL>  select snap_id, INSTANCE_NUMBER,PARAMETER_NAME,VALUE from dba_hist_parameter where
PARAMETER_NAME='cpu_count' order by snap_id;

   SNAP_ID INSTANCE_NUMBER PARAMETER_NAME                                                  VALUE
---------- --------------- ---------------------------------------------------------------- ------
. . . . . .
      3368               1 cpu_count                                                        48
      3369               1 cpu_count                                                        48
      3369               2 cpu_count                                                        48
      3370               1 cpu_count                                                        48
      3371               1 cpu_count                                                        48
      3372               1 cpu_count                                                        48
      3373               1 cpu_count                                                        48
      3374               1 cpu_count                                                        48
      3375               2 cpu_count                                                        96
      3375               1 cpu_count                                                        48
      3376               1 cpu_count                                                        48
      3376               2 cpu_count                                                        96
      3377               1 cpu_count                                                        48
      3377               2 cpu_count                                                        96
      3378               2 cpu_count                                                        96
      3378               1 cpu_count                                                        48
      3379               1 cpu_count                                                        48
      3379               2 cpu_count                                                        96
. . . . . .
```

查询告警日志：more alert*|grep -i Cpu 也可以获取 CPU 的变化。

询问客户可知，是他们调整过系统的 CPU 资源，所以导致节点 2 上的 CPU_COUNT 自动变化，引起了 enq: IV - contention 等待。

若主机的 CPU 个数变化，那么当主机重启后数据库的 cpu_count 参数的值会随之变化，该参数属于操作系统依赖参数。

调整主机的 CPU 个数之后，该等待事件消失。

1.4 MOS

1.4.1 12c RAC DDL Performance Issue: High "enq: IV - contention" etc if CPU Count is Different (文档 ID 2028503.1)

12c RAC DDL Performance Issue High enq IV - contention etc if CPU Count is Different (文档 ID 2028503.1).mhtml