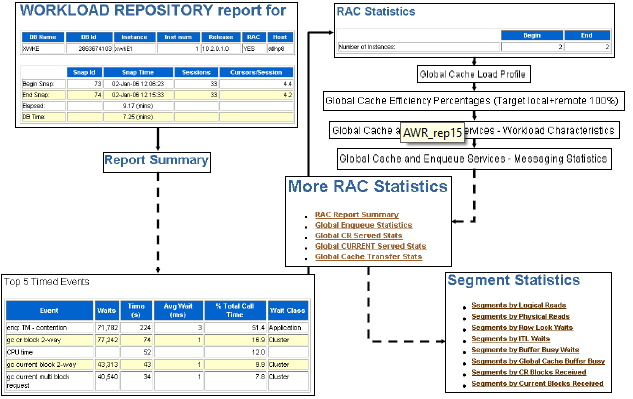
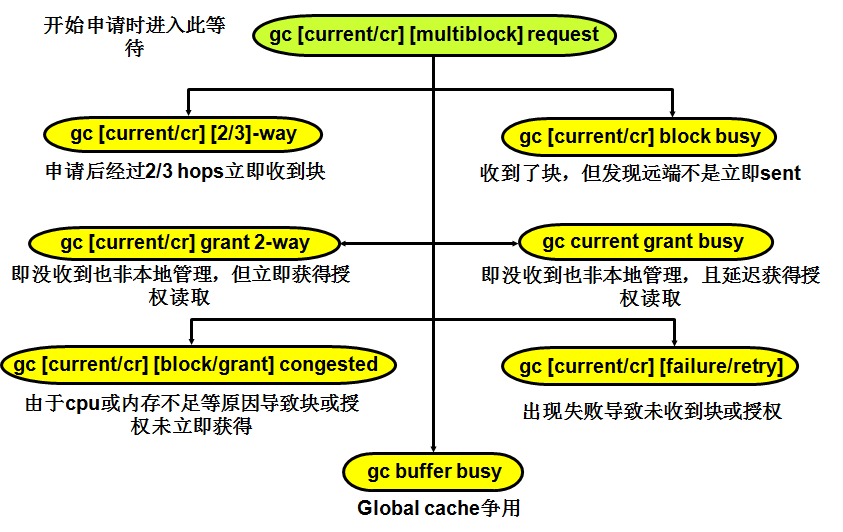
>AWR RAC 特定环节

RAC cache fusion  我为人人，人人为我

预备知识-纵览 RAC AWR

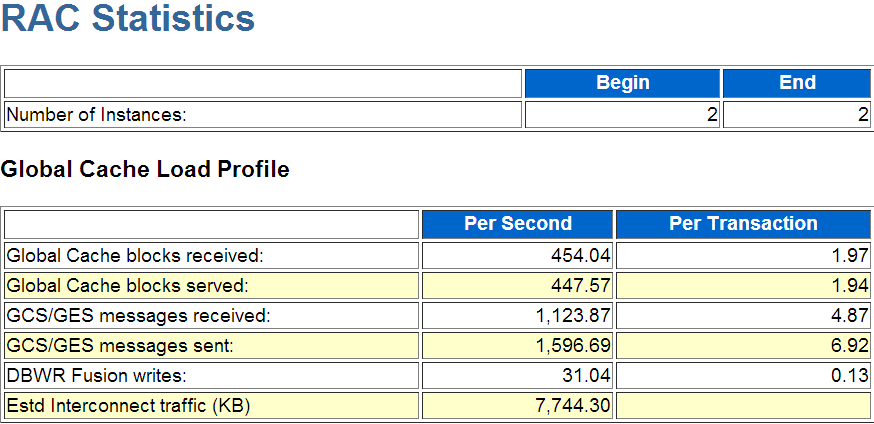


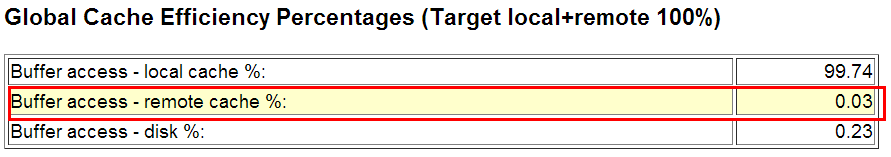
预备知识-Global Cache等待事件



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)





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

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

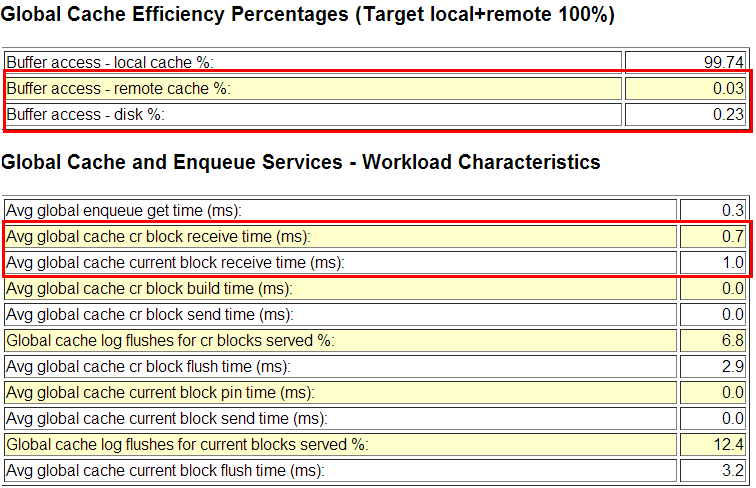
b)网络延迟

**>EM** 监控 **Cluster Interconnect**

• 比你手动去查效率高

• 并可定制interconnect 告警

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



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 J !

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!!

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

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

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

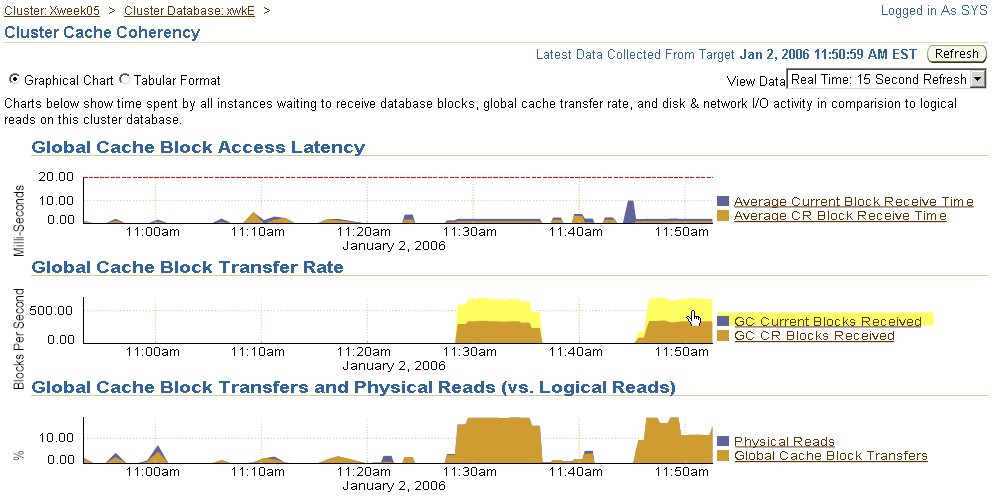
Time to process current block request in the cache= (pin time + flush time + send time) 相关指标：

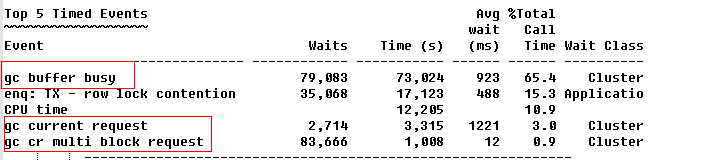
•gc current block pin time

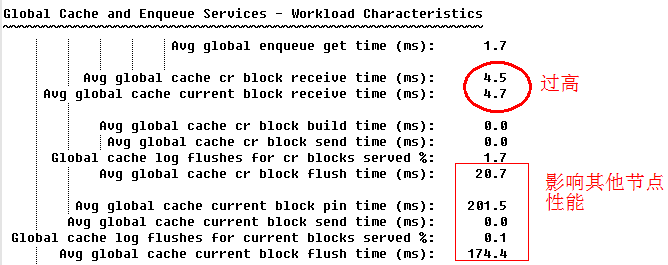
•gc current block flush time

•gc current block send time

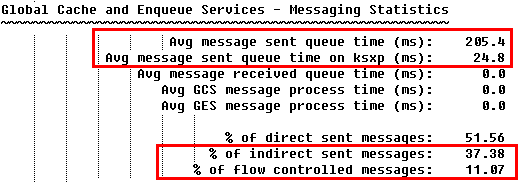
Cluster Cache Coherency







Avg global cr/current block receive time高>è 本实例出现大量cluster等待 Avg global cache pin/send/flush time高>è 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



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 è 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

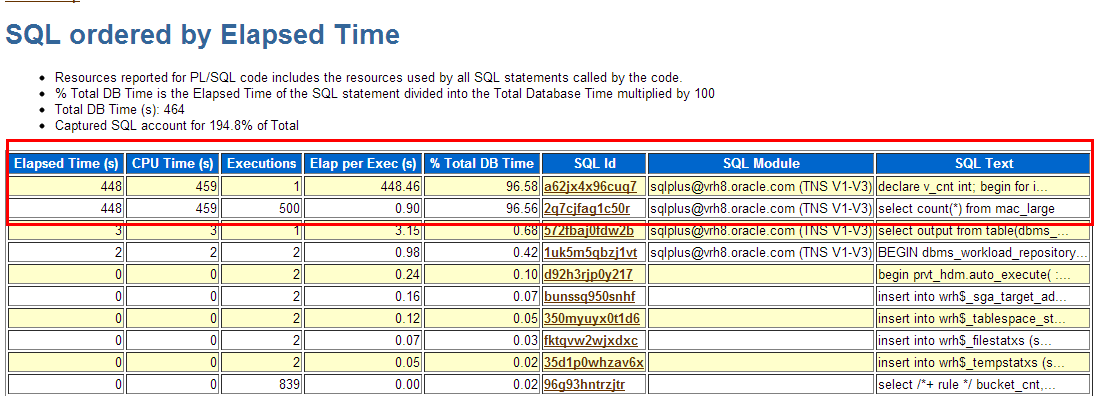
% of flow controlled messages è 流控制最常见的原因是网络状况不佳， % of flow controlled messages应当小于1%

....

....

....

**>AWR SQL Statistics**



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。

IMG_266

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

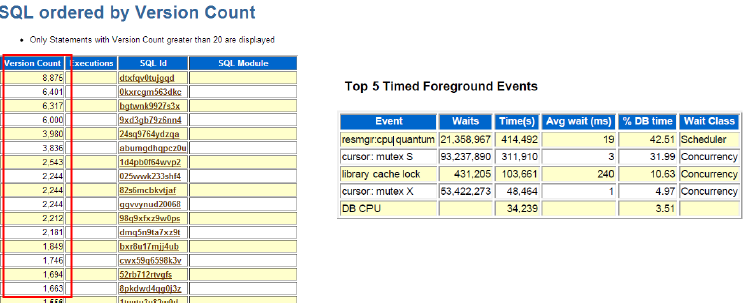
1.%Total - Elapsed Time as a percentage of Total DB time è这个语句耗总的 DB TIME的百分之多少

2.%CPU - CPU Time as a percentage of Elapsed Time è 这个语句耗费的 DB TIME里 CPU占多少比例 è这个语句是否是 CPU敏感的

3.%IO - User I/O Time as a percentage of Elapsed Time è 这个语句耗费的 DB TIME里IO 占多少比例è这个语句是否是 IO敏感的

4.Captured SQL account for 94.8% of Total DB Time (s) è 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



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。