

ParnassusData is a software company

V\$SESSTAT 和 V\$SYSSTAT 统计信息描述

Parnassus
诗 檀

诗檀软件 www.parnassusdata.com

工程师：王通
报告生成日期：2016 年 7 月 13 日
更新日期：2016 年 7 月 13 日



文档控制

变更记录

日期	作者及更新人	版本号	变更信息
2016-07-13	王通	1.0	Initial

审阅人

版本号	审阅人	职位	相关评论
1.0	刘相兵	技术经理	

审批人

版本号	批准人	日期	相关评论

文档分发

分发号	文档名	分发位置
1		

级别	意思
1	User
2	Redo
4	Enqueue
8	Cache
16	OS
32	Real Application Clusters
64	SQL
128	Debug

名字	级别	备注 2
background checkpoints completed	8	后台进程完成检查点的个数。
background checkpoints started	8	后台进程开始的检查点数量。如果一个新的检查点覆盖了未完成的检查点或者这个检查点还在运行,可能会比"background checkpoints completed"大,这个统计只包括 redo 的检查点,不包括: offline 和 begin backup 这些操作产生的独立文件检查点; Foreground (用户请求)检查点 (例如 ALTER SYSTEM CHECKPOINT LOCAL 语句等)。
branch node splits	128	由于插入了额外的数据导致索引分支块分裂的数量。
buffer is not pinned count	72	访问缓冲区时 free 的次数。仅限于内部 debug 使用。
buffer is pinned count	72	访问时缓冲区 pin 的次数。仅限于内部 debug 使用。
bytes received via SQL*Net from client	1	通过 Oracle Net Services 从客户端收到的字节总数。
bytes received via SQL*Net from dblink	1	通过 Oracle Net Services 从 dblink 收到的字节总数。
bytes sent via SQL*Net to client	1	通过前台进程发送给客户端的字节总数。
bytes sent via SQL*Net to dblink	1	通过 dblink 发送的字节总数。
Cached Commit SCN referenced	128	这个统计仅限内部 debug 为目的来使用。
calls to get snapshot scn:kcmgss	32	SCN 分配的次数。SCN 号在事务开始时分配。
calls to kcmgas	128	获得一个新的 SCN, 调用 kcmgas 的次数。
calls to kcmgcs	128	获得一个当前的 SCN, 调用 kcmgcs 的次数。
calls to kcmgrs	128	获得一个最近的 SCN, 调用 kcmgrs 的次数。
change write time	8	当前块的变化被写入 redo 的时间, 单位 10 毫秒。
cleanouts and rollbacks - consistent read gets	128	块回滚和块清除时要求一致性读的次数。请参考后述的"consistent gets"。
cleanouts only - consistent read gets	128	块清除要求的一致性读次数, 不包括回滚。请参考后述的"consistent gets"。
cluster key scan block gets	64	集群扫描获得数据块的数量。
cluster key scans	64	集群扫描开始的次数。
cold recycle records	8	recycle cache 使用 LRU 算法读的缓冲区的数量。
commit cleanout failures:block lost	8	ORACLE 准备 commit 块清除时, 不能找到正确的块的次数。
commit cleanout	8	ORACLE 准备 commit 块清除时, 缓冲区正在被写入的次数。

failures:buffer being written		
commit cleanout failures:callback failure	8	ORACLE 准备 commit 块清除时, 调用 callback 函数返回 false 的次数。
commit cleanout failures:cannot pin	8	ORACLE 准备 commit 块清除时, 块不能被 pin 的次数。
commit cleanout failures:hot backup in progress	8	ORACLE 准备 commit 块清除时, 在热备期间的次数。块需要被记录在缓冲区变脏之前。
commit cleanout failures:write disabled	8	ORACLE 准备 commit 块清除时, 数据库写操作被暂时关闭的次数。
commit cleanouts	8	commit 执行时, 块清除的总次数。
commit cleanouts successfully completed	8	commit 执行时, 块清除成功的总次数。
Commit SCN cached	128	commit 操作的系统变更号被缓冲的次数。
consistent changes	8	用户进程使用 rollback 保证块一致性读的次数。工作负载产生大量的一致性变化会消耗大量的资源。这个统计值的大小应该比"consistent gets" 小。
consistent gets	8	块的一致性读请求次数。请参考"consistent changes"和"session logical reads"。
CPU used by this session	1	一个 call 从发起到结束, 花费的 CPU time (单位 10ms) 数量。如果一个 call 不足 10ms, 那么统计的时候按照 0 计算。在操作系统的报告中也会出现一个相似的问题, 尤其是系统遭受多次环境变更的情况下。
CPU used when call started	128	启动 call 时, 所花费的 CPU 时间。请参考之前所说的"CPU used by this session"。
CR blocks created	8	当前块被克隆成 CR 块的数量。克隆最常见的原因是缓冲区处于非兼容性模式。
current blocks converted for CR	8	当前块被转化成 CR 状态的数量。
cursor authentications	128	操作执行时管理执行权限的次数。
data blocks consistent reads - undo records applied	128	为了读一致性, 应用于数据块的 undo 记录次数。
db block changes	8	和"consistent changes"关系紧密, 这个统计记录了 SGA 中对所有的块的 update 和 delete 操作总数。如果事务被提交, 这样的变化会产生 redo 日志因此永久改变数据库。这近似所有的数据库工作, 这个统计表明缓冲区正在变脏的比率 (基于每个事务或者每秒)
db block gets	8	当前块请求的次数。请参考之前的 consistent gets 。
DBWR buffers scanned	8	ORACLE 扫描 LRU 链时脏块和干净块的数量, 除以"DBWR lru scans"得到平均每次扫描数据块的数量。
DBWR checkpoints buffers written	8	检查点时, dbwr 写入的脏块数量。
DBWR checkpoints	8	DBWR 被请求扫描缓存, 给所有的数据块标记检查点和完成恢复的次数。这个统计经常比"background checkpoints completed" 要大。
DBWR cross instance writes	40	仅限 RAC 环境: 满足另外的实例的锁, 被写入的块的数量。
DBWR free buffers found	8	当 DBWR 进程被要求释放空缓冲区时, 干净的缓冲区的数量。除以 "DBWR make free requests" 可以得到 LRU 链的重用缓冲区平均数量。
DBWR lru scans	8	DBWR 扫描 LRU 链寻找缓冲区写的次数。包括扫描一个满的队列为了另外的目的 (例如, checkpoint) 。这个统计中经常比"DBWR make free requests"要大, 或者相等。
DBWR make free requests	8	DBWR 进程使 LRU 链释放一些缓冲区的请求次数。
DBWR revisited being-written buffer	8	DBWR 尝试保存一个正在写的缓冲区并发现它已经处于写的批次的次数。这个统计值统计了 DBWR 尝试填满一个批次做的"无用"工作的数量。

DBWR summed scan depth	8	DBWR 每次扫描 LRU 的脏缓冲区，都会增加当前扫描深度（DBWR 检查缓冲区的数量）这个统计值。除以"DBWR lru scans"可以得到平均扫描深度。
DBWR transaction table writes	8	回滚段头被 DBWR 写入的数量。这个统计值表明了有多少“热”缓冲区被写入，导致用户进程等待当写完成时。
DBWR undo block writes	8	DBWR 进程写的回滚段数据块数量。
DDL statements parallelized	32	并行执行的 DDL 语句的数量。
deferred (CURRENT) block cleanout applications	128	延时块清除的次数。
DFO trees parallelized	32	串行执行计划变更成并行计划的次数。
dirty buffers inspected	8	当用户进程寻找缓冲区重用，找到的脏缓冲区数量。
DML statements parallelized	32	并行执行的 DML 语句的数量。
enqueue conversions	4	表或者行锁转换的次数。
enqueue deadlocks	4	不同会话中表或者行锁死锁的次数。
enqueue releases	4	表或者行锁释放的次数。
enqueue requests	4	表或行锁申请的次数。
enqueue timeouts	4	表或者行锁（申请和转换）在完成前超时的次数。
enqueue waits	4	因为队列获取延迟导致发生在队列转换或获取期间的等待时间
exchange deadlocks	8	当两个缓冲区交换并出现一个内部重启错误时，进程发现潜在死锁的次数。索引扫描是导致这种交换的唯一因素
execute count	64	调用（用户和递归）执行 sql 语句。
free buffer inspected	8	扫描 LRU 队列找可重用缓冲区跳过的缓冲区个数。同"dirty buffers inspected"的差异在于一些缓冲区不可用，因为它们正在被用户和等待者读或写，或着在更新之后处于忙碌或者正在被写的状态。
free buffer requested	8	重用缓冲区或空的缓冲区被要求创建或负载一个数据块的次数。
global cache blocks corrupt	40	仅限 RAC：数据块发生错误或链接校验错误的次数
global cache convert time	40	仅限 RAC：锁转换花费的时间
global cache convert timeouts	40	global cache 超时锁转换的时间
global cache converts	40	global cache 中锁转换的次数
global cache cr block receive time	40	前台进程通过 interconnect 等待发送 CR 数据块的时间，这个统计值除以 "global cache cr blocks received" = 每个数据块等的时间
global cache cr block serve time	40	BSP 进程构建 CR 块的时间。这个指标值除以"global cache cr blocks served"= 每个 CR 块构建时间
global cache cr blocks received	40	收到的块总数。
global cache cr blocks served	40	BSP 进程所构造的块总数。
global cache cr requests blocked	40	前台进程请求 CR 块失败的次数
global cache cr timeouts	40	前台进程请求 CR 块时超时的次数
global cache defers	40	锁被请求并且锁的拥有者延迟释放锁的次数
global cache freelist waits	40	系统配置更少的 lock element 和 buffers 比。前台进程等待 lock element 的次数
global cache get time	40	总的等待时间。这个值除以"global cache gets"= 每个请求的平均等待时间

global cache gets	40	获得的锁定数。
global cache cr block send time	40	BSP 进程发生构造 CR 数据块的时间。这个统计值除以"global cache cr blocks served"= 每个 CR 块的发送时间
global cache cr block log flushes	40	日志刷新一致性读块的数量
global cache cr block log flush time	40	在构造一个 CR 块之后, BSP 进程日志刷新花费的时间。这个统计值除以 "global cache cr blocks served" = 每个 CR 块的日志刷新时间
global cache prepare failures	40	Interconnect transfer 准备时发生故障的次数。
global lock async converts	32	异步全局锁转换的次数
global lock async gets	32	异步全局锁获取的次数
global lock convert time	32	同步和异步全局锁转换时间, 单位 10ms
global lock get time	32	同步和异步全局锁获取时间, 单位 10ms
global lock releases	32	同步全局锁释放的数量
global lock sync converts	32	同步全局锁转换的数量
global lock sync gets	32	同步全局锁获取的数量
hot buffers moved to head of LRU	8	当一个热块到达 LRU 队列的尾部的时候, ORACLE 自动会把这个块移动到 LRU 队列的头上使其可以重用。这个统计像这样的移动次数
immediate (CR) block cleanout applications	128	在一致性读请求期间, 立即应用清除记录的次数
immediate (CURRENT) block cleanout applications	128	在 current gets 期间, 立即应用清除记录的次数。对照"deferred (CURRENT) block cleanout applications"
index fast full scans (direct read)	64	使用直接读入来启动的高速全盘扫描的次数。
index fast full scans (full)	64	对段全体启动的高速全扫描的次数。
index fast full scans (rowid ranges)	64	指定的 ROWID endpoint 启动的高速全扫描的次数。
instance recovery database freeze count	32	实例恢复时被冻结的次数。
kcmccs called get current scn	32	当需要确认 SCN 号时, kernel 获得当前 SCN 的次数。
kcmgss read scn without going to DLM	32	没有分布锁管理 (DLM), kernel 获得 SCN 的次数。
kcmgss waited for batching	32	数据库进程阻塞等待 SCN 的次数。
leaf node splits	128	因为插入额外的值导致索引叶节点分裂的次数
logons cumulative	1	实例启动后, 登录的次数。只在 V\$SYSSTAT 有用, 它提供了所有登录实例的进程。
logons current	1	目前登录的数量。只在 V\$SYSSTAT 有用
messages received	128	后台进程间收到的消息数量
messages sent	128	后台进程间发送的消息数量
native hash arithmetic execute	64	执行哈希操作使用本身算法而不是 ORACLE NUMBERS 的次数
native hash arithmetic fail	64	执行哈希操作使用本身算法失败, 而要求使用 ORACLE NUMBERS 的次数
no buffer to keep pinned count	72	尝试访问一个缓冲区, 但是缓冲区没有被找到的次数。像 "buffer is not pinned count" 和 "buffer is pinned count", 这个统计只是用来内部 debug 使用。
no work-consistent read gets	128	既不请求块清除也不请求回滚的 consistent gets 次数, 请参考前述的 consistent gets。

opened cursors cumulative	1	V\$SYSSTAT : 实例启动后, 开启的游标数量。 V\$SESSTAT : 会话开始后, 开启的游标数量。
opened cursors current	1	目前开启的游标总数。
opens of replaced files	8	文件因为不在文件缓冲中而不得不重新打开的次数。
opens requiring cache replacement	8	打开文件导致在文件缓存中的当前文件被关闭的次数。
OS All other sleep time	16	操作系统 Sleep 时间, 除去 "OS Data page fault sleep time", "OS Kernel page fault sleep time", "Text page fault sleep time", "OS User lock wait sleep time".
OS Chars read and written	16	操作系统读和写的字节数。
OS Data page fault sleep time	16	由于丢失数据段导致的 sleep 时间。
OS Input blocks	16	I/O 读的数量。
OS Involuntary context switches	16	操作系统强制切换环境的次数。
OS Kernel page fault sleep time	16	由于操作系统 kernel page 错误导致的 sleep 时间。
OS Major page faults	16	导致 I/O 的 page faults 数量。
OS Messages received	16	接收的信息数量。
OS Messages sent	16	发送的页面错误的数量。
OS Minor page faults	16	I/O 没执行的页面错误的数量。
OS Other system trap CPU time	16	处理系统缺陷 (作为独特的系统调用) 的总时间。
OS Output blocks	16	写入 I/O 的数量。
OS Process heap size	16	内存中被进程分配的区域大小。典型的方式是 malloc () 获得内存。
OS Process stack size	16	进程 stack segment 的大小。
OS Signals received	16	接收的信号的数量。
OS Swaps	16	swap page 的数量。
OS System call CPU time	16	执行系统模式花费的 sleep 时间。
OS System calls	16	system call 的数量。
OS Text page fault sleep time	16	由于缺失文本段花费的 sleep 时间
OS User level CPU time	16	执行用户模式花费的 sleep 时间。
OS User lock wait sleep time	16	等待操作系统对象锁花费的 sleep 时间。
OS Voluntary context switches	16	自动 context 切换的次数 (例如, 当一个进程释放 CPU 通过一个 SLEEP() 系统调用)。
OS Wait-cpu (latency) time	16	等待 CPU 变的可用花费的 sleep 时间。
Parallel operations downgraded 1 to 25 pct	32	并发执行被请求但因为并发执行服务器不足导致并行度减少的次数。
Parallel operations downgraded 25 to 50 pct	32	并发执行被请求但因为并发执行服务器不足导致并行度减少的次数。
Parallel operations downgraded 50 to 75 pct	32	并发执行被请求但因为并发执行服务器不足导致并行度减少的次数。
Parallel operations downgraded 75 to 99 pct	32	并发执行被请求但因为并发执行服务器不足导致并行度减少的次数。
Parallel operations downgraded to serial	32	并发执行被请求但因为并发执行服务器不足导致串行执行的次数。
Parallel operations not downgraded	32	并发执行被执行在请求的并发度上的次数。

parse count (hard)	64	解析 calls (真解析) 的次数。一次硬解析是非常昂贵的操作对内存来说, 因为它需要 ORACLE 分配工作堆和内存结构去建立一个解析树。
parse count (total)	64	解析 calls (硬解析和软件析) 的总次数。一次软解析是检查对象是否在共享池中, 保证下面的对象权限没有改变。
parse time cpu	64	用于解析的 CPU 总时间, 单位 10ms。
parse time elapsed	64	解析花费的总时间, 单位 10ms。减去 "parse time cpu" 得到等待解析资源的时间。
physical reads	8	数据块从磁盘读的次数。这个值比 "physical reads direct" 加上 "physical reads cache" 大一些。
physical reads direct	8	不通过缓存直接从磁盘读的数量。
physical writes	8	数据块写入磁盘的总数。这个统计值等于 "physical writes direct" 加 "physical writes from cache"。
physical writes direct	8	直接写入磁盘, 不通过缓冲区缓存的数量。
physical writes non checkpoint	8	非 CHECKPOINT 引起的物理写的次数。
pinned buffers inspected	8	当一个用户进程扫描 REPLACEMENT 列表, 寻找一个可重用的 BUFFER 的时候, 发现一个冷块被 PIN 了或者有一个 PIN 请求的等待事件。这种情况很少发生, 因为冷块很少会被 PIN。
prefetched blocks	8	连续或者不连续的被预读的数据块的数量。
prefetched blocks aged out before use	8	连续或者不连续的被预读但在用之前老化的数据块的数量。
process last non-idle time	128	这个块最后被执行的时间。
PX local messages rcv'd	32	current 会话的本地实例中, 用于并行执行的接收的本地信息的数量。
PX local messages sent	32	current 会话的本地实例中, 用于并行执行的发送的本地信息的数量。
PX remote messages rcv'd	32	current 会话的本地实例中, 用于并行执行的接收的远程消息的数量。
PX remote messages sent	32	current 会话的本地实例中, 用于并行执行的发送的远程信息的数量。
queries parallelized	32	并行执行的 SELECT 语句的数量。
recovery array read time	8	恢复中 I/O 的耗时间。
recovery array reads	8	恢复中被执行的读取数。
recovery blocks read	8	恢复中被读取的块数。
recursive calls	1	用户和系统级别的递归调用数量。ORACLE 维持表用于内部进程。当 ORACLE 需要对这些表做出改变时, 它内部会产生一个内部 sql 语句, 轮流产生递归调用。
recursive cpu usage	1	CPU time 被用于非用户调用 (递归调用) 的时间。减去 "CPU used by this session" 得到多少 cpu time 用于用户调用。
redo buffer allocation retries	2	重新申请分配 redo buffer 空间的次数。重申因为 redo write 未完成或者日志切换发生错误。
redo entries	2	redo 信息被拷贝到 redo log buffer 的次数。
redo log space requests	2	当前 ACTIVE 的日志文件满了, Oracle 必须等待日志切换完成后才能分配 REDO LOG 磁盘空间。 日志文件大小和 SGA 不匹配, Commit 提交频繁会产生问题。当日志切换的时, ORACLE 需要确认所有提交的脏缓冲区都被写入磁盘在切换到一个新的日志文件之前。如果你有一个充满脏缓冲区的大 SGA 和一个小的日志文件, 日志切换必须等 DBWR 将脏缓冲区写入到磁盘上之后。
redo log space wait time	2	等待 "redo log space requests" 的总时间, 单位 10ms。
redo log switch interrupts	2	另外实例请求这个实例切到下一个日志文件。
redo ordering marks	2	一次强制性的 redo 记录产生的 scn 更大和使用相同数据块的其他线程产生的 scn 相比。
redo size	2	redo 产生的总字节数。
redo synch time	8	"redo synch writes" 耗费的时间, 单位 10ms。

redo synch writes	8	应用于 log buffer 的变更必须被写入磁盘的次数, 取决于一次 commit。一般来说 redo 信息生成后, 会被拷贝到 log buffer 中, 并不需要马上被写入 redo log 文件, lgwr 会周期性将这些数据写入 redo log 文件。
redo wastage	2	redo 数据块会被写入, 在 redo 数据块完成之前浪费的字节数。
redo write time	2	从日志缓冲区写入到日志文件的时间, 单位 10ms。
redo writer latching time	2	LGWR 获得每个 copy latch 的时间, 单位 10ms。
redo writes	2	LGWR 写入日志文件的数量。"redo blocks written"除以这个统计值等于每次写的数据库块数量。
remote instance undo block writes	40	当前实例写 rollback segment 以至于另外的实例可以读的次数。
remote instance undo header writes	40	当前实例写 undo header block 以至于另外的实例可以读的次数。
rollback changes-undo records applied	128	用户请求回滚 (不是 consistent-read 回滚) 是撤销的记录数量。
rollbacks only-consistent read gets	128	consistent gets 请求块回滚的次数, 不包括块清除。请参考"consistent gets"。
rows fetched via callback	64	CALLBACK 函数返回的记录数。该统计量仅仅用于内部 DEBUG。
serializable aborts	1	一个 sql 语句在处于串行的隔离级别不得不终止的次数。
session connect time	1	会话连接时间, 单位 10ms。只在 V\$SESSTAT 有用。从登录会话开始算时间。
session cursor cache count	64	总的游标缓存数量。这个统计值增长除非 SESSION_CACHED_CURSORS > 0。这个统计值在 V\$SESSTAT 中最有用。V\$SESSTAT 中的这个统计值接近于 SESSION_CACHED_CURSORS 参数的设定值, 应该增加参数值。
session cursor cache hits	64	会话游标缓存的命中数量。一次命中意味着 sql 不需要再次解析。减去 "parse count (total)" 得到实际解析数。
session logical reads	1	"db block gets"加上"consistent gets"的值。包括缓冲区数据块的逻辑读和进程私有内存的数据块的逻辑读。
session pga memory	1	会话的当前 PGA 大小。这个值仅在 V\$SESSTAT 时有效。在 V\$SYSSTAT 中没意义。
session pga memory max	1	会话的最大 PGA 大小。这个值仅在 V\$SESSTAT 时有效。在 V\$SYSSTAT 中没意义。
session stored procedure space	1	会话中用于存储过程的内存大小。
session uga memory	1	会话的当前 UGA 值。这个值仅在 V\$SESSTAT 时有效。在 V\$SYSSTAT 中没意义。
session uga memory max	1	会话的最大 UGA 值。这个值仅在 V\$SESSTAT 时有效。在 V\$SYSSTAT 中没意义。
sorts (disk)	64	至少一次磁盘写入的排序操作次数。磁盘上排序很消耗 I/O, 尝试增加参数 "SORT_AREA_SIZE"。更多信息, 参考"SORT_AREA_SIZE"。
sorts (memory)	64	在内存中完成不需要任何磁盘写入的排序操作次数。
sorts (rows)	64	排序行的总数。
SQL*Net roundtrips to/from client	1	Net8 通过客户端发送和接收的信息总数。
SQL*Net roundtrips to/from dblink	1	Net8 通过 dblink 发送和接收的信息总数。
summed dirty queue length	8	每次写请求之后脏数据 LRU 队列长度总和。写完成之后除以写请求次数得到平均队列长度。
switch current to new buffer	8	当前块移动到不同的缓冲区, 留下一个 CR 块在原来的缓冲区, 这种操作的次数。
table fetch by rowid	64	通过 rowid 访问表的次数 (一般是通过索引)。这一般发生在表扫描的时候, 说明不是最优的查询, 或者没有索引。
table fetch continued row	64	访问一个表遇到行链接和行迁移的次数。

table scan blocks gotten	64	在扫描期间, oracle 按照顺序检索每一行。这个统计记录在扫描期间能被发现的块的数量。这个统计告诉你为了扫描从缓冲区得到的数据库块的个数。和 consistent gets 对比来判断 consistent read 的活跃性是否归功于扫描。
table scan rows gotten	64	在扫描期间, 被处理的行的数量。
table scans (cache partitions)	64	对 cache 可用的表执行范围扫描的次数。
table scans (direct read)	64	执行直接读的全表扫描的次数 (绕过 buffer cache)
table scans (long tables)	64	长表被定义为不满足短表定义的表, 短表定义参考下文 "table scans (short tables)"。
table scans (rowid ranges)	64	并行查询中, 表扫描使用指定 ROWID 范围的次数。
total file opens	8	实例打开文件的总数。
transaction lock background get time	128	这个统计仅在以内部 debug 为目的时, 可以使用。
transaction lock background gets	128	这个统计仅在以内部 debug 为目的时, 可以使用。
transaction lock foreground requests	128	这个统计仅在以内部 debug 为目的时, 可以使用。
transaction lock foreground wait time	128	这个统计仅在以内部 debug 为目的时, 可以使用。
transaction rollbacks	128	成功被回滚的事务数。
transaction tables consistent read rollbacks	128	回滚段头被回滚生成一致性读块的次数。
transaction tables consistent reads - undo records applied	128	以一致性读为目的的 undo 信息应用于事务表进行回滚的次数。
Unnecessary process cleanup for SCN batching	32	因为会话或进程没得到下一批 SCN 而不必要执行进程清除的次数。下一批 SCN 被另一个会话代替。
user calls	1	用户 call 的次数, 例如 login、parse、fetch、or execute。
user commits	1	用户提交 commits 的次数。用户提交一个事务时, redo 产生数据库块的变动的映射必须被写入磁盘。Commits 通常代表用户事务的比率。
user rollbacks	1	用户手动执行 rollback 语句的次数或者在用户的事务期间发生的错误次数。
write clones created in background	8	前台或者后台进程克隆一个当前正在被写入的缓冲区的次数。克隆出来的称为新的, 可用的缓冲区, 离开原来的缓冲区 (现在是克隆的) 完成写入。
write clones created in foreground	8	前台或者后台进程克隆一个当前正在被写入的缓冲区的次数。克隆出来的称为新的, 可用的缓冲区, 离开原来的缓冲区 (现在是克隆的) 完成写入。

其他问题

未解决的问题

问题号	问题描述	解决方案	日期

已解决的问题

问题号	问题描述	解决方案	解决日期

诗檀软件Oracle技术团队，国内最专业的Oracle数据库服务，<http://www.parnassusdata.com>



ParnassusData Corporation, Shanghai, GaoPing Road No. 733. China

Phone: (+86) 400-690-3643

ParnassusData.com

Copyright © 2013, ParnassusData and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, or including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0410

Copyright © 2016ParnassusData Corporation. All Rights Reserved.