In-Memory-Undo in Oracle 10g-11g

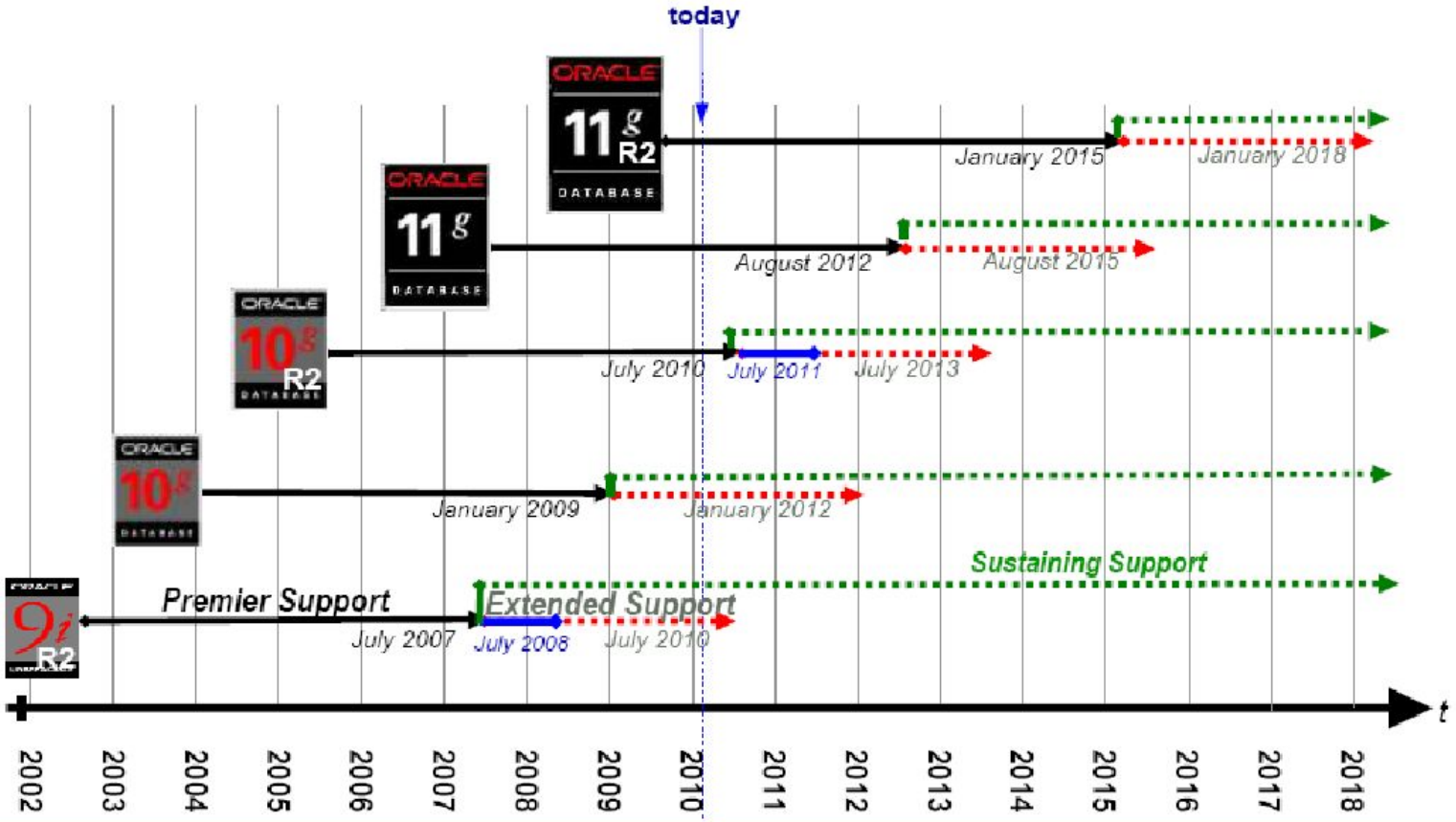
Ю.Пудовченко

Открытые Технологии

Сведения об IMU отсутствуют в:

- Oracle Database 10R1g Documentation
- Oracle Database 11gR1 Documentation
- Oracle Database 11gR2 Documentation
- MyOracleSupport

Lifetime Support Policy



ORACLE

Как это бывает:

- **Bug 9406607 Corrupt blocks in 11.2 in table with unique key.**
- **Applies to:** Oracle Server - Enterprise Edition - Version: 11.2.0.1 to 11.2.0.1 - Release: 11.2 to 11.2
  Information in this document applies to any platform.

- **Description:** Block corruption during recovery for a table with Primary or Unique key constraint and lots of inserts done on that table due to Bug 9406607 .

  The recovery can happen as part of normal STARTUP during crash recovery.
- **Workaround or Resolution:** In order to avoid this problem set parameter _in_memory_undo = FALSE (Disable IMU).

  To repair the existent corruption, the table may need to be recovered from a backup or use DBMS_REPAIR to skip the affected blocks.

  The redo being applied from the redo log is already corrupt meaning that regular media recovery does not fix the corruption.

Bug 9406607: CORRUPT BLOCKS DURING RECOVERY. ORA-600 [KDBLKCHECKERROR] WHEN CHECKING ENABLED

- **Type:** B – Defect
- **Fixed in Product Version:** 12.1
- **Severity**1: Complete Loss of Service
- **Product Version:** 11.2.0.1.0
- **Platform:** IBM AIX on POWER Systems (64-bit)
- **Created:** 24-Feb-2010
- **Updated:** 31-Mar-2010
- **Affects Platforms:** Generic

ORA-04031

ORA-00474: SMON process terminated with error
Instance terminated by PMON, pid = 2448

startup open

IMODE=BR
ILAT =363

Completed: ALTER DATABASE OPEN

Doing block recovery for file 4 block 1091148
Block recovery from logseq 20655, block 22669 to scn 30108486586

Recovery of Online Redo Log: Thread 1 Group 2 Seq 20655 Reading mem 0
  Mem# 0: /oradata/redo02.log
Block recovery completed at rba 20655.24427.16, scn 7.43715515

Errors in file /oradata/admin/bdump/orcl_smon_25757.trc:
ORA-00600: internal error code, arguments: [ktbair1], [4], [1], [], [], [], [], []

Errors in file /oradata/admin/bdump/orcl_pmon_25739.trc:
ORA-00474: SMON process terminated with error
PMON: terminating instance due to error 474
Instance terminated by PMON, pid = 25739

IMU

- В Shared Pool создается область для хранения информации отката Undo Pool по принципу «Один пул- одна транзакция».
- Место в сегменте отката резервируется, заголовок сегмента отката обновляется, но информация отката направляется в IMU pool.
- Польза: транзакция выполняется быстрее.
- Каждый пул защищается собственной защелкой «in memory undo»
- IMU pool = 65535 bytes
- IMU pool выделяются в порядке LRU
- _imu_pools = 10% от TRANSACTIONS

IMU

- Условия для работы IMU:
- compatibility >= 10.0.
- undo_management= AUTO

Преимущества IMU

- Для IMU уменьшается редо. Несколько redo record записываются одной операцией I/O экономия в/в. уменьшение 'redo wastage'.
- Несколько обновлений одного UNDO-блока объединяются в одно изменение и записываются на диск одной операцией.
- IMU находится в памяти, поэтому consistent reads выполняемые другими сессиями завершаются быстрее. экономия ЦПУ

Особенности

- ITL в блоке данных стал указывать на IMU вместо _SYSSMU

- IMU enabled by default

- Starting ORACLE instance (normal)
- …
- **IMODE=BR - Batched Redo Mode**
- **ILAT =27 - Number of IMU Pools/Latches**
- …
- Starting up:
- Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production

Особенности (-)

- IMU не работает в RAC. т.е. как только CLUSTER_DATABASE=TRUE…
- Logminer can't handle all new structures in redologs yet
- + logical standby and streams
- Bug 9406607 Corrupt blocks in 11.2 in table with unique key
  ORA-600 [kdblkcheckerror] [ID 1078406.1]

8 параметров управляют IMU-ом

_in_memory_undo = TRUE|FALSE - включает/отключает IMU, def.=TRUE.

_imu_pools - Кол-во IMU-пулов

PARAMETER 11.1	SESS_VALUE	INST_VALUE
_db_writer_flush_imu	TRUE	TRUE
_recursive_imu_transactions	FALSE	FALSE
_imu_pools	3	3

PARAMETER 11.2	SESS_VALUE	INST_VALUE
_db_writer_flush_imu	TRUE	TRUE
_recursive_imu_transactions	FALSE	FALSE
_imu_pools	3	3

8 параметров управляют IMU-ом

- PROCESSES - Чем больше это значение, тем больше пулов и IMU latches.
- Shalahamer: 8.25 processes/one IMU latch
- Haisley: 10%*Transactions

Согласно документации:Transactions=1.1*Sessions, Sessions=1.1*Processes+5

Следовательно: PROC=300 => (1,1*1,1*300+5)/10= 368

	PROC	TRAN	ILAT	Init/ILAT
11.1	300	368	36	8,3
11.2	300	519	51	6

11.1 transactions	368
11.2 transactions	519

show parameter cpu

cpu_count	4
parallel_threads_per_cpu	2
resource_manager_cpu_allocation	4

8 параметров управляют IMU-ом

- **_recursive_imu_transactions** – Использовать IMU для рекурсивных транзакций. The default is FALSE.
- **_db_writer_flush_imu** – «allows Oracle the freedom to artificially age a transaction for increased automatic cache management» def.=TRUE.

IMU Pools

- Функции для выделения/освобождения IMU в коде СУБД:

- ktiImuPoolAllocate
- ktiImuPoolFree
- ktiNonImuPoolAllocate
- ktiNonImuPoolFree
- ktiNonImuPoolLatchClean
- kti_disable_imu
- kti_enable_imu
- kti_is_imu
- ktiimu_chg
- ktiimu_f
- ktiredo_imupoolfree
- ktiundo_imupoolallocate

- IMU-пулы можно видеть через представление X$ktifp. Размер каждого пула 65535 байт:

SQL> select ktifpno,ktifppsi from x$ktifp;

KTIFPNO	KTIFPPSI
0	65535
1	3
2	4
3	65535
4	65535
5	65535

KTIFPNO	KTIFPPSI
0	2
1	3
2	65535
3	65535
…	

KTIFPNO	KTIFPPSI
4	65535
5	4
6	65535
…	

Транзакции

- обновляется заголовок undo-сегмента
- в undo-сегменте выделяется место под данный IMU-пул,
- физическая запись из IMU в блок UNDO-сегмента происходит несколько позже (при наступлении commit или flush - при заполнении пула до 100%). Эта запись производится одной операцией (batch)
- блоки, содержащие информацию отката X$BH.FLAG: 0x8002 (PRIVATE CURRENT)

Запись на диск

- Max changes exceeded (~100)
- Contention flushes (write induced)
- Bitmap state change flushes
- Rollback flushes
- Commit flushes
- Multi-block undo flushes (undo > blksize)

Запись на диск

KTIFFCAT	KTIFFFLC	
Undo pool overflow flushes	**25300**	<-- Переполнение undo-пула
Stack cv flushes	23448	
Multi-block undo flushes	**0**	
Max. chgs flushes	**10197**	<-- более 100 изменений
NTP flushes	0	
Contention flushes	**19727**	<-- Contention
Redo pool overflow flushes	**385**	<-- Переполнение redo-пула
Logfile space flushes	0	
Multiple persistent buffer flushes	0	
Bind time flushes	0	
Rollback flushes	355467	
Commit flushes	**2105449**	<-- «стандартных» commit
Recursive txn flushes	**79628**	<-- рекурсивные транзакции
Redo only CR flushes	174	
Ditributed txn flushes	**0**	
Set txn use rbs flushes	0	
Bitmap state change flushes	69772	
Presumed commit violation	0	
Securefile direct-write lob update flushes	**0**	

IMU Latches

- select name, gets, misses, immediate_gets IM, sleeps
- from v$latch_children where name like 'In%undo%'

Instance Parameters	GETS	MISSES	IM	SLEEPS
In-memory undo latch	0	0	0	0
In-memory undo latch	0	0	0	0
In-memory undo latch	7623	0	3815	0
In-memory undo latch	33806	0	17437	0
In-memory undo latch	2919	0	478	0
In-memory undo latch	21624	0	356	0

IMU Locks

```
select * from v$lock_type where type='IM'

TYPE NAME            ID1_TAG ID2_TAG IS_USER
---- --------------- ------- ------- -------
IM   Kti blr lock pool #             0 NO
```

DESCRIPTION

Serializes block recovery for an IMU txn

Статистики

SQL> select name from v$statname where name like 'IMU%';

IMU commits
IMU Flushes
IMU contention
IMU recursive-transaction flush
IMU undo retention flush
IMU ktichg flush
IMU bind flushes
IMU mbu flush
IMU pool not allocated
IMU CR rollbacks
IMU undo allocation size
IMU Redo allocation size
IMU- failed to get a private strand

IMU Commits

select * from v$sysstat where name like '%commit%'

STAT#	NAME	CLASS	VALUE	STAT_ID
5	**user commits**	1	120366	582481098
312	**IMU commits**	128	98992	1914489094

Мониторинг

```
SQL> select * from v$sgastat
     where name like 'KTI%';
11.1
POOL         NAME                            BYTES
------------ -------------------- -------
shared pool KTI latch structure     5760
shared pool KTI-UNDO                2535552
shared pool KTI pool states           40
shared pool KTI latches             1152
11.2 □       KTI freelists            88
```

Выброшенное UNDO

UNDO = DATA + REDO

Conventional	Throw Away Undo
REDO redo.log UNDO redo.log, undo.dbf	REDO redo.log UNDO undo.dbf

Throw Away UNDO

О его включении оповещает нас строки в alert.log:

…

IMODE=TUA <- Batched Redo mode

ILAT=308 <- Number of IMU pools/latches

…

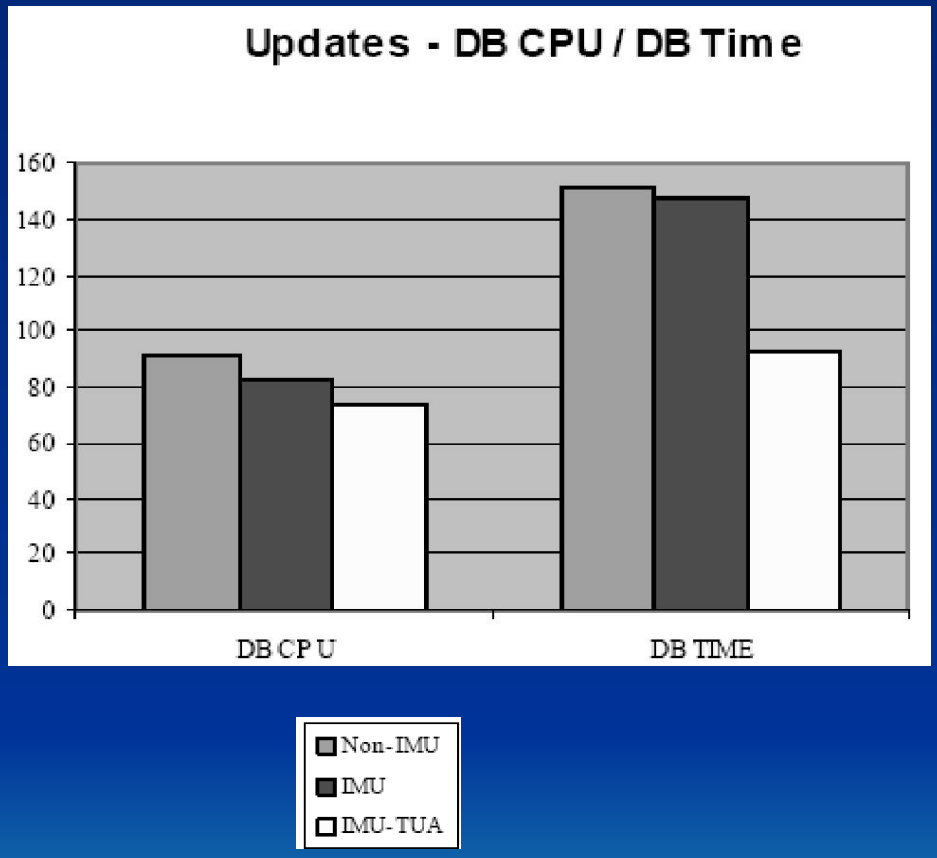
Events

- ORA-30047 Internal event for kti tracing
- ORA-30048 Internal event for IMU autotuning

Производительность

- Craig Shalahamer:
- Benefit: около 20% ЦПУ
- If you check v$sesstat, you will see the redo size is reduced. I did a test before, the redo size in v$sesstat of a single row update was reduced by 17% (the percentage depends on your DML).

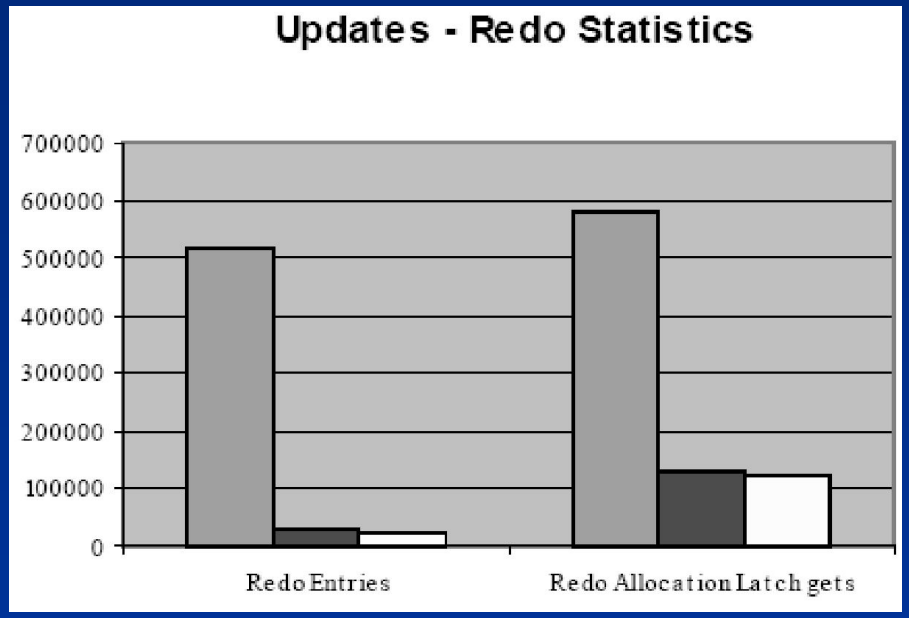
Haisley



Updates

	DB CPU	DB Time
IMU:	9%	19%
TAU:	19%	39%

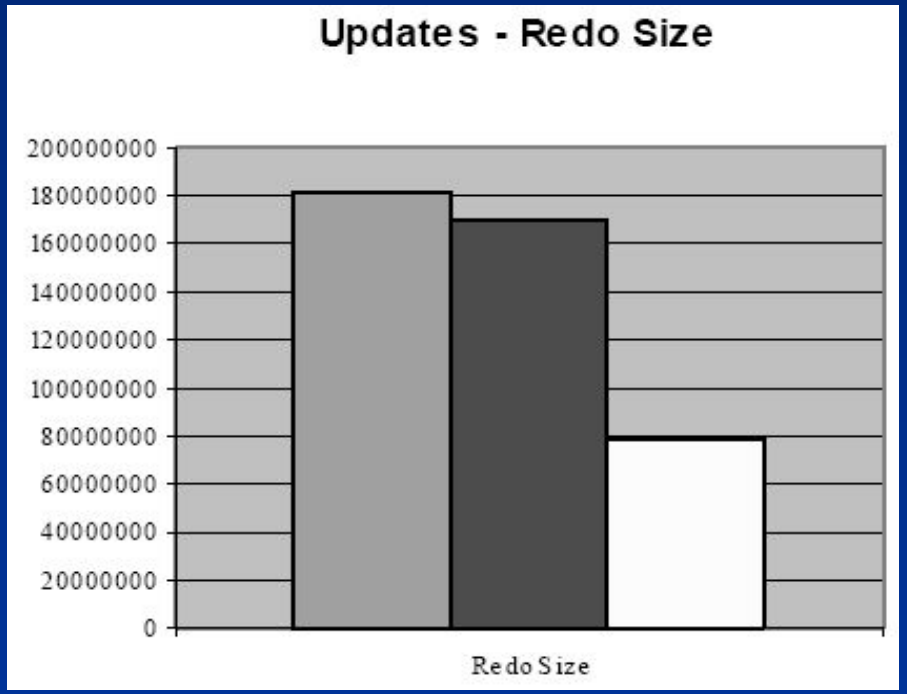
Haisley



Redo Statistics

	Entries	Защелка RAL
IMU:	94%	77%
TAU:	96%	79%

Haisley



Redo Size

	Redo Size
IMU:	6,5%
TAU:	57%

Литература

- Craig A. Shallahamer, All About Oracle’s In-Memory Undo
- Stephan Haisley, Center Of Expertise, Oracle Corporation