

Nologging Objects 11g – 18c

Timo Giese, DOAG 2018, 21.11.18

Agenda

- 1 About Me**
- 2 About Us
- 3 Overview
- 4 Nologging Operations
- 5 Prevent Nologging Operations
- 6 Detect Nologging Operations
- 7 "Repair" Corrupted Objects
- 8 Logging vs Nologging Datablock
- 9 12.2 Improvements
- 10 18c Improvements
- 11 Flashback Database and Datapump

About Me

- Oracle DBA since 2005
- Oracle High Availability Specialist
- Ansible and Vagrant Enthusiast
- Twitter: @mbe7



Agenda

- 1 About Me
- 2 About Us**
- 3 Overview
- 4 Nologging Operations
- 5 Prevent Nologging Operations
- 6 Detect Nologging Operations
- 7 "Repair" Corrupted Objects
- 8 Logging vs Nologging Datablock
- 9 12.2 Improvements
- 10 18c Improvements
- 11 Flashback Database and Datapump

Diverse Target Markets

More than 1,100 customers and owners in four target markets are the basis of our successful business strategy.



Focusing on customers in 4 target markets

We provide services to customers in four target markets – Our focus is on our owners in cooperative banking - Volksbanken und Raiffeisenbanken

Cooperative Financial Network*

Cooperative Banks:
Volksbanken
Raiffeisenbanken



Founding principle

Companies within the
Cooperative Financial
Network



Integration promoters

Cooperative Specialized
Commercial Banks



One IT in the CFN

Market Customers



Focused Market
Expansion

Our Locations

Five locations in Germany ensure high security and best customer contact.

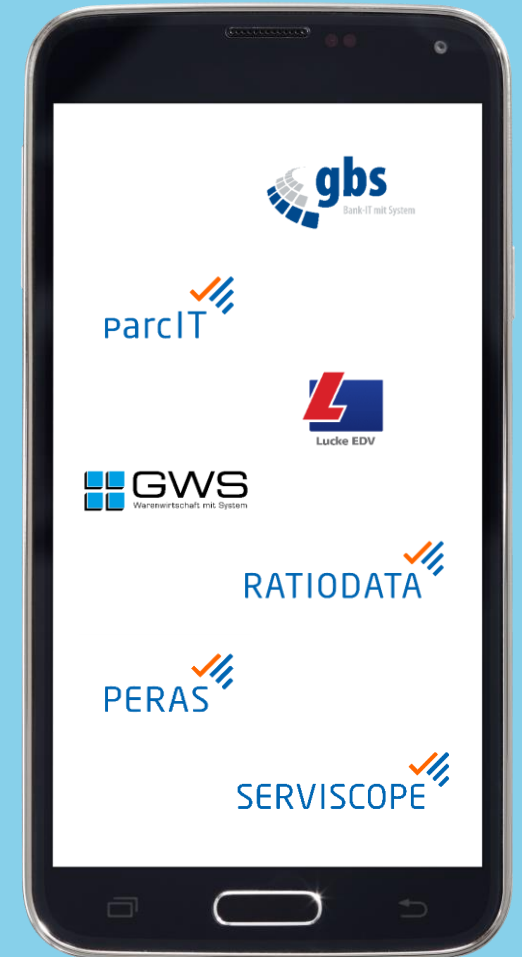
- **Karlsruhe and Münster**
Board, Administration, Product Management, Development, Production, Sales, Customer Service and Printing
- **München**
Product Management, Development, Sales and Customer Service
- **Berlin**
Sales and Printing
- **Frankfurt**
Registered Office and Sales



Our Corporate Group

The Group companies offer a wide range of IT services.

- Systems and services for HR departments
- Specialized and process IT consulting for private and special banks, planning of bank-specific solutions
- Software solutions for risk management and controlling
- On-site technical infrastructure services
- Consulting, project management, software development, quality assurance
- Product management for cooperation-related business and service providers
- Business Process Outsourcing



Computer Center Performance

The performance of our four computer centers – More than just zeros and ones.

Performance



Virtual servers: **48,930**

Nine mainframe computers: **196,733 MIPS**

Mainframe transactions: **56,800m per year**

Network data volumes at computer centers: **approx. 3.17 Petabyte per day**

Utilized cartridge capacity: **62,757 Terabyte**

Mainframe storage: **3,164 Terabyte** / Open system storage: **23,833 Terabyte**

Agenda

- 1 About Me
- 2 About Us
- 3 Overview**
- 4 Nologging Operations
- 5 Prevent Nologging Operations
- 6 Detect Nologging Operations
- 7 "Repair" Corrupted Objects
- 8 Logging vs Nologging Datablock
- 9 12.2 Improvements
- 10 18c Improvements
- 11 Flashback Database and Datapump

Overview

What are Nologging Objects?

- minimally logged transactions
- redolog is generated (no full transaction redolog record)
- leads to unrecoverable datafiles
- Full- or incremental Backup is needed after nologging operation
- could lead to Oracle error 26040 / 1578 "Data block was loaded using the NOLOGGING option"
- after recovery datafiles could be corrupt and data needs to be inserted again through application

Overview

Use Cases

- Data Warehouse
- DSS
- Mixed Workload systems
- large amount of data
- staging areas
- time critical - data has to be processed in a short time

Overview

Nologging Objects

- Tables (Table Data)
- Table Partitions
- Indices
- Lob Objects (Table Data)

Agenda

- 1 About Me
- 2 About Us
- 3 Overview
- 4 Nologging Operations**
- 5 Prevent Nologging Operations
- 6 Detect Nologging Operations
- 7 "Repair" Corrupted Objects
- 8 Logging vs Nologging Datablock
- 9 12.2 Improvements
- 10 18c Improvements
- 11 Flashback Database and Datapump

Nologging Operations

- insert /*+ append */ into
- insert into <table> nologging
- create table <table_copy> as select
- alter table <table> move|split partition nologging
- alter index <index> rebuild nologging
- alter index <index> split|rebuild partition nologging
- SQL-Loader direct load (sqlldr direct=true)

Agenda

- 1 About Me
- 2 About Us
- 3 Overview
- 4 Nologging Operations
- 5 Prevent Nologging Operations**
- 6 Detect Nologging Operations
- 7 "Repair" Corrupted Objects
- 8 Logging vs Nologging Datablock
- 9 12.2 Improvements
- 10 18c Improvements
- 11 Flashback Database and Datapump

Prevent Nologging Operations

- Performance vs Recovery
 - general policy (consistency vs performance)
 - Dataguard Systems*
-
- globally disabled
 - *alter database force logging;*
 - per tablespace
 - *alter tablespace <tbs> force logging;*
-
- turn off globally
 - *alter database no force logging;*
 - turn off per tablespace
 - *alter tablespace <tbs> no force logging;*

Prevent Nologging Operations

Is force logging enabled?

- *select force_logging from v\$database;*

```
SYS@ORCL SQL > select force_logging from v$database;
```

```
FORCE_LOGGING
```

```
-----  
YES
```

- *select tablespace_name,force_logging from dba_tablespaces;*

```
SYS@ORCL SQL > select tablespace_name,force_logging from dba_tablespaces;
```

```
TABLESPACE_NAME    FORCE_LOGGING
```

```
-----  
TBS1                NO  
TBS2                YES  
TBS3                NO
```

- *select table_name,logging from dba_tables;*

```
SYS@ORCL SQL > select table_name,logging from dba_tables;
```

```
TABLE_NAME          LOGGING
```

```
-----  
table1              NO  
table2              YES  
table3              YES
```

Agenda

- 1 About Me
- 2 About Us
- 3 Overview
- 4 Nologging Operations
- 5 Prevent Nologging Operations
- 6 Detect Nologging Operations**
- 7 "Repair" Corrupted Objects
- 8 Logging vs Nologging Datablock
- 9 12.2 Improvements
- 10 18c Improvements
- 11 Flashback Database and Datapump

Detect Nologging Operations

- DB-Verify Utility (NONCDB): dbv
 - generates message 'DBV-00200' and 'DBV-00201' as result for nonlogged blocks
 - *dbv file='<path_to_df>/file.dbf' userid=sys/pw*
- DB >= 10.2.0.5 with RMAN
 - *validate check logical database;*
 - *validate check logical datafile 3;*
 - 11g:
 - View *v\$database_block_corruption* displays "CORRUPTION_TYPE=NOLOGGING"
 - 12c:
 - View: *v\$nonlogged_block*
- DB >= 11g
 - *report unrecoverable;*
- DB >= 12.2 RMAN
 - validate database nonlogged block;
 - validate datafile nonlogged block;

Detect Nologging Operations

- *RMAN validate*

```
RMAN> validate check logical datafile 10;
```

```
Starting validate at 2018.11.11 - 15:38:37
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=435 device type=DISK
channel ORA_DISK_1: starting validation of datafile
channel ORA_DISK_1: specifying datafile(s) for validation
input datafile file number=00010 name=+DATA/ORCL/65404F36A8F4572AE0540208201005DE/DATAFILE/test.281.991926729
channel ORA_DISK_1: validation complete, elapsed time: 00:00:01
```

List of Datafiles

=====

File	Status	Marked	Corrupt	Empty	Blocks	Blocks	Examined	High	SCN
10	OK	1308			9989	12800		18647556	
File Name: +DATA/ORCL/65404F36A8F4572AE0540208201005DE/DATAFILE/test.281.991926729									
	Block Type	Blocks	Failing	Blocks	Processed				
	Data	0		1313					
	Index	0		0					
	Other	0		1498					

```
Finished validate at 2018.11.11 - 15:38:41
```

Detect Nologging Operations

- *RMAN report unrecoverable (>=11g)*

```
RMAN> report unrecoverable;

using target database control file instead of recovery catalog
Report of files that need backup due to unrecoverable operations
File Type of Backup Required Name
-----
10 full or incremental +DATA/ORCL/65404F36A8F4572AE0540208201005DE/DATAFILE/test.281.991926729
```

- *similar output to rman "report unrecoverable"*

```
SELECT df.name "datafile", df.unrecoverable_time
FROM v$datafile df, v$backup bk
WHERE df.file#=bk.file#
and df.unrecoverable_change#!=0;
```

datafile	UNRECOVERABLE_TIME
+DATA/CTSTDBX2/65404F36A8F4572AE0540208201005DE/DATAFILE/test.281.991929727	2018.11.11 - 16:05:29

Detect Nologging Operations

- *show objects defined as nologging:*

```
select distinct ss.owner,ss.object_name, ss.object_type,
ss.tablespace_name, ts.logging tablespace_level_logging
from v$segment_statistics ss, dba_tablespaces ts,v$datafile df
where ss.statistic_name ='physical writes direct'
and ss.value >0 and df.unrecoverable_change# >0
and ss.ts#=df.ts# and ss.tablespace_name=ts.tablespace_name;
```

```
OWNER OBJECT_NAME OBJECT_TYPE TABLESPACE_NAME TABLESPACE_LEVEL_LOGGING
-----
SYS    T3              TABLE      TEST              NOLOGGING
```

- *v\$nonlogged_block (12c)*

```
select file#,block#,blocks,NONLOGGED_START_CHANGE#,NONLOGGED_END_CHANGE# from v$nonlogged_block where rownum <=1;
```

```
FILE#    BLOCK#    BLOCKS  NONLOGGED_START_CHANGE#  NONLOGGED_END_CHANGE#
-----
10       267      13      18647507                 18647507
```

Agenda

- 1 About Me
- 2 About Us
- 3 Overview
- 4 Nologging Operations
- 5 Prevent Nologging Operations
- 6 Detect Nologging Operations
- 7 "Repair" Corrupted Objects**
- 8 Logging vs Nologging Datablock
- 9 12.2 Improvements
- 10 18c Improvements
- 11 Flashback Database and Datapump

“Repair” Nologging Objects

- Indices
 - rebuild index (*alter index <idx> rebuild online;*)
- Tables/Table Partitions
 - data in blocks are lost and have to be inserted again from the application
 - if it is a free block, block will be overwritten, when write requested in the future

- use package DBMS_REPAIR
 - first step: skip corrupt blocks for table
 - second step: move table

```
BEGIN
DBMS_REPAIR.SKIP_CORRUPT_BLOCKS (
SCHEMA_NAME => '&schema_name',
OBJECT_NAME => '&table_name',
OBJECT_TYPE => dbms_repair.table_object,
FLAGS => dbms_repair.SKIP_FLAG);
END;
/

alter table &table_name move [online];
```

- further information: MOS-Note 794505.1

“Repair” Nologging Objects

- LOBs
 - further details in MOS-Note 293515.1

- Dataguard Environment
 - Standby documented in MOS-Note 958181.1 (Rolling a Standby Forward using RMAN Incremental Backup To Fix The Nologging Changes)
 - DB \geq 12c in MOS-Note 95818.1 (Rolling Forward a Physical Standby Using Recover From Service Command in 12c)

Demo

Agenda

- 1 About Me
- 2 About Us
- 3 Overview
- 4 Nologging Operations
- 5 Prevent Nologging Operations
- 6 Detect Nologging Operations
- 7 "Repair" Corrupted Objects
- 8 Logging vs Nologging Datablock**
- 9 12.2 Improvements
- 10 18c Improvements
- 11 Flashback Database and Datapump

Logging vs Nologging Datablock

- alter system dump datafile x block y;

```
Start dump data blocks tsu: 6 file#:14 minblk 189 maxblk 189.← Tbs File-ID
Block dump from disk: ← Block read from disk
buffer tsu: 6 rdba: 0x038000bd (14/189)
scn: 0x4a4773 seq: 0x01 flg: 0x06 tail: 0x47730601
fmt: 0x02 chkval: 0x6196 type: 0x06=trans data
Hex dump of block: st=0, typ found=1
Dump of memory from 0x00007F0B86EE7000 to 0x00007F0B86EE9000
7F0B86EE7000 0000A206 038000BD 004A4773 06010000 [.....sGJ.....]
...
7F0B86EE8FE0 00000003 0202012C 570503C1 646C726F [.....World]
7F0B86EE8FF0 0202012C 480502C1 6F6C6C65 47730601 [,.....Hello..sG]
Block header dump: 0x038000bd
Object id on Block? Y
seg/obj: 0x12f7b csc: 0x00000000004a476d itc: 2 flg: E typ: 1 - DATA
brn: 0 bdba: 0x38000b8 ver: 0x01 otc: 0
inc: 0 exflg: 0

Itl          Xid          Uba          Flag Lck          Scn/Fsc
0x01 0x0001.018.00000748 0x02403cab.0232.0d --U- 2 fsc 0x0000.004a4773
0x02 0x0000.000.00000000 0x00000000.0000.00 ---- 0 fsc 0x0000.00000000
bdba: 0x038000bd
data block dump data header at 0x7f0b86ee7064
=====
tsiz: 0x1f98
hsiz: 0x16
pbl: 0x7f0b86ee7064
76543210
```

```
flag=-----
ntab=1
nrow=2
fxxx=-1
fsbc=0x16
fseq=0x1f80
symp=0x1f6a
tosp=0x1f6a
0xe:pti[0] nrow=2 offs=0
0x12:pri[0] offs=0x1f8c
0x14:pri[1] offs=0x1f80
block row dump: ← rows and column values
tab 0, row 0, @0x1f8c
tl: 12 fb: --H-FL-- lb: 0x1 cc: 2
col 0: [ 2] c1 02
col 1: [ 5] 48 65 6c 6c 6f
tab 0, row 1, @0x1f80 ← row 1
tl: 12 fb: --H-FL-- lb: 0x1 cc: 2
col 0: [ 2] c1 03 ← column 1 ASCII-Character
col 1: [ 5] 57 6f 72 6c 64 ← column 2 ASCII-Character
end of block dump
End dump data blocks tsu: 6 file#: 14 minblk 189 maxblk 189
```

Logging vs Nologging Datablock

- *alter system dump logfile '<path>/redo.log' layer 19;*

– layer 19: nologging changes

```
...
REDO RECORD - Thread:1 RBA: 0x000095.00000690.0060 LEN: 0x003c VLD: 0x01 CON_UID: 1919301561
SCN: 0x000000000269014 SUBSCN: 1 11/16/2018 11:50:23
(LWN RBA: 0x000095.0000068c.0010 LEN: 0x00000005 NST: 0x0001 SCN: 0x000000000269013)
CHANGE #1 INVLN CON_ID:3 AFN:60 DBA:0x0f000091 BLKS:0x000f OBJ:73571 SCN:0x000000000269014 SEQ:1 OP:19.2 ENC:0 FLG:0x0000
Direct Loader invalidate block range redo entry
...
```

```
select owner,object_name,object_id from dba_objects where object_id=73571;
```

OWNER	OBJECT_NAME	OBJECT_ID
TEST	T_NOLOGGING	73571

Logging vs Nologging Datablock

- block is corrupt after recovery and will be skipped when trying to dump its content
 - „*table scan: segment: file# 13 block# 2258 skipping corrupt block file# 13 block# 2259*“
- every query which includes this datablock won't be successful
 - *“ORA-26040: Data block was loaded using the NOLOGGING option”*
 - Trace Event can mitigate this issue and select will show the “good” data
 - *alter session set events='10231 trace name context forever';*

Agenda

- 1 About Me
- 2 About Us
- 3 Overview
- 4 Nologging Operations
- 5 Prevent Nologging Operations
- 6 Detect Nologging Operations
- 7 "Repair" Corrupted Objects
- 8 Logging vs Nologging Datablock
- 9 12.2 Improvements**
- 10 18c Improvements
- 11 Flashback Database and Datapump

12.2 Improvements

- Recover nonlogged blocks on Dataguard Standby database
 - requires Active Dataguard License
 - Nonlogged Block list from `v$nonlogged_block` is populated to the Standby Database controlfile
 - recover with RMAN on Standby
 - first disable redo apply (*edit database set state=apply-off*)
 - *recover database nonlogged block;*
 - reenable redo apply (*edit database set state=apply-on*)

```
RMAN> recover database nonlogged block;
Starting recover at 24-OCT-18
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=22 device type=DISK
```

```
starting recovery of nonlogged blocks
List of Datafiles
=====
File Status Nonlogged Blocks Blocks Examined Blocks Skipped
-----
1 OK 0 0 232474
2 OK 0 0 35231
3 OK 0 0 84758
4 OK 0 0 48462
7 OK 0 12531 20563
```

```
Details of nonlogged blocks can be queried from v$nonlogged_block view
recovery of nonlogged blocks complete, elapsed time: 00:00:15
```

Agenda

- 1 About Me
- 2 About Us
- 3 Overview
- 4 Nologging Operations
- 5 Prevent Nologging Operations
- 6 Detect Nologging Operations
- 7 "Repair" Corrupted Objects
- 8 Logging vs Nologging Datablock
- 9 12.2 Improvements
- 10 18c Improvements**
- 11 Flashback Database and Datapump

18c Improvements

- 2 new modes
 - Standby Nologging for Data Availability
 - Standby Nologging for Load Performance
- Active Dataguard License needed!!!
- Only supported on Exadata and Oracle Cloud!!!
(<https://docs.oracle.com/en/database/oracle/oracle-database/18/dblic/Licensing-Information.html#GUID-0F9EB85D-4610-4EDF-89C2-4916A0E7AC87>)
- Standby Nologging for Data Availability
 - dedicated connection to all standbys
 - commit delayed until changes are applied on all standbys
- Standby Nologging for Load Performance
 - all nologging data is directly sent to standby databases
 - if not possible due to network bottlenecks, nologging data transport will be stopped. Same situation as on primary for nologging operations. Datafile changes will be applied to Standby during normal managed recovery apply.

Agenda

- 1 About Me
- 2 About Us
- 3 Overview
- 4 Nologging Operations
- 5 Prevent Nologging Operations
- 6 Detect Nologging Operations
- 7 "Repair" Corrupted Objects
- 8 Logging vs Nologging Datablock
- 9 12.2 Improvements
- 10 18c Improvements
- 11 Flashback Database and Datapump**

Flashback Database and Datapump

- What can Flashback Database do for nologging operations?
 - create restore point before nologging operation
 - do nologging operations
 - in case of any issues, go back to restore point and do operation again
- DB >= 12.1
 - speedup datapump import
 - during import objects will be set to nologging and at the end back to logging
 - TRANSFORM=DISABLE ARCHIVE LOGGING:Y
 - restrict to a subset
 - TRANSFORM=DISABLE_ARCHIVE_LOGGING:Y:TABLE

**Thank you very much for
your attention!**

