

About the Processes in Oracle

“Andernfalls Prozess am Hals”

Eero Mattila

Principal Systems Consultant
Dell Software

eero.mattila@software.dell.com



Agenda

1. Introduction
2. Brief historical review
3. Process definitions
4. What are the background processes doing?
5. Multithreaded execution in Oracle12c – why bother?

Who am I?

- Started with Oracle V6 in 1991
 - DBA, Forms/Reports Developer
- Oracle Germany 1995
 - RDBMS, Forms, Reports, Designer
- Quest / Dell Software since 2005
 - Oracle and SQL Server Tools – Toad, Spotlight, SQL Optimizer
 - Database Replication – SharePlex
 - Performance Monitoring – Foglight



Brief historical review

- Oracle7
 - Lots of new features (referential integrity, stored procs, triggers, etc.)
 - A few new background processes (CKPT, RECO, LCKn, SNPn) – but not mandatory
 - Ran with a minimum of 4: DBWR, LGWR, PMON, SMON
- Oracle8i and 9i
 - Java, AQ, RAC
 - Few more processes, still easy to handle
- Oracle10g
 - ASM, Data Guard
 - 35 different background processes in all
- Oracle11g
 - New memory management
 - 50 new background process types
- Oracle12c
 - ILM, Multitenant Database
 - 400+ background processes in V\$BGPROCESS



Process definitions

- In this presentation, Process = Thread
 - Windows: one executable, one process, many threads
 - Unix/Linux: one executable, many processes
- Server processes
 - Handle requests from user sessions
 - Often the most intensive resource consumers in the database instance
- Background processes
 - Start up with the instance
 - Focussed on specific tasks
 - Some are mandatory, others optional, depending on installed features (RAC, ASM, DG, AQ etc.)
- Slave or Worker processes
 - Spawned by either Server or Background processes to perform special tasks



Server processes

- Start when a user connects to the database
- Perform all work based on clients' requests: execute SQL, join, sort, sum – whatever
- Two types: Dedicated Server and Shared Server
- Dedicated Server
 - 1-to-1 relationship between client connection and server process
 - 100 client connections => 100 server processes
 - Each process has its own isolated address area, no session will block another
 - Default, and recommended, connection method in Oracle
 - Necessary for certain operations, hence always automatically set up
 - No special configuration necessary



Server processes - 2

- Shared Server (fka Multithreaded Server, MTS)
 - 1-to-many relationship between server process and client connections
 - E.g. 1,000 client connections, 10 Shared Servers => 10 server processes instead of 1,000
- Client connects to a Dispatcher process
- Dispatcher assigns a free Shared Server to the client session
- No dedicated server-to-session mapping -> one session can execute three statements with three different shared servers
- When used correctly, potential for memory savings and performance gain
 - 10 processes instead of 1,000 => 990 PGAs less (but more SGA memory needed)
- Caution: Only suitable for short transactions, i.e. OLTP systems
 - "Monopolization" of shared servers may cause other sessions to wait
 - 2 shared servers, 3 sessions -> all 3 run a 30 sec transaction simultaneously
-> the first two get their response in 30 seconds, the third in a minute!
 - "Artificial deadlocks" possible



Server processes - 3

- To configure Shared Servers

1. `alter system set shared_servers = n scope = both; -- n > 0`
2. Connect string or TNSNAMES entry must include the magic word SHARED
`ora12centos2:1521/DOAG:SHARED`

```
DOAG_SHARED=  
  (DESCRIPTION=  
    (ADDRESS=(PROTOCOL=TCP) (HOST=ora12centos2) (PORT=1521))  
    (CONNECT_DATA=  
      (SERVER=SHARED)  
      (SERVICE_NAME=DOAG)  
    )  
  )  
)
```



Server processes - 4

- Database Resident Connection Pooling (DRCP) since Oracle11g
 - Combination of dedicated and shared server concepts
- A pre-allocated pool of shared servers
- Once a server is assigned to a session, the session owns the server process until it disconnects
- Designed for APIs that don't have their own connection pool, eg. PHP
- Suitable for systems with large numbers of users, short and frequent connections
- Pooling benefits of shared servers, performance characteristics of dedicated servers, no "artificial deadlocks"
- Configure with `DBMS_CONNECTION_POOL`
Connect string or `TNSNAMES` entry needs the word `POOLED`



Server processes – conclusions

- As long as your server is not overloaded, and there is no other compelling reason to use a shared server configuration, go for dedicated server
- No additional configuration for dedicated server
- Shared server sessions are more complicated to tune
 - Data spread over several trace files
- If you have to use shared server, make sure you develop and TEST in a shared server environment



Background and Slave Processes

- Oracle Documentation on the one hand, and the Data Dictionary on the other, make it hard to distinguish between Background, Slave and Worker processes.
- Oracle 12c Database Reference Guide, Appendix F:
 - "a **background process** is defined as any process that is listed in V\$PROCESS and has a non-null value in the PNAME column."

```
select count(*) from v$process where pname is not null;
```

```
COUNT (*)  
-----  
55
```



Background and Slave Processes – 2

- The View `v$PROCESS` lists all running processes within an Oracle instance. The column `BACKGROUND` has the value 1 for all background processes.

```
select count(*) from v$process where background = 1;
```

```
COUNT (*)
```

```
-----
```

```
32
```



Background and Slave Processes – 3

- The View **v\$BGPROCESS** contains an entry for all background processes that can be started in an Oracle instance. Currently running processes have a value other than '00' in the column PADDR.

```
select count(*) from v$bgprocess where paddr <> '00';
```

```
COUNT (*)  
-----  
          22
```

- Needless to say: All queries were executed immediately one after another.
- Hmm.



Background and Slave Processes – 4

- The load on the database did not change all of a sudden.
- Oracle simply don't seem to agree with themselves:
 - The documentation considers all slave and worker processes as backgrounds, all in all some 400 processes of around 110 types.
 - V\$BGPROCESS lists 70+ process types for a database with a minimal configuration. It contains some, but not all utility processes, and no slaves or worker processes.
 - V\$PROCESS shows all running processes of the instance. It has its very own opinion on which are background processes and which are not. It recognizes e.g. Space Management Slaves, but not Parallel Query Slaves, or Job Queue Processes.
- We shall stick to the documentation: All processes not directly related to user sessions, are considered as background processes.

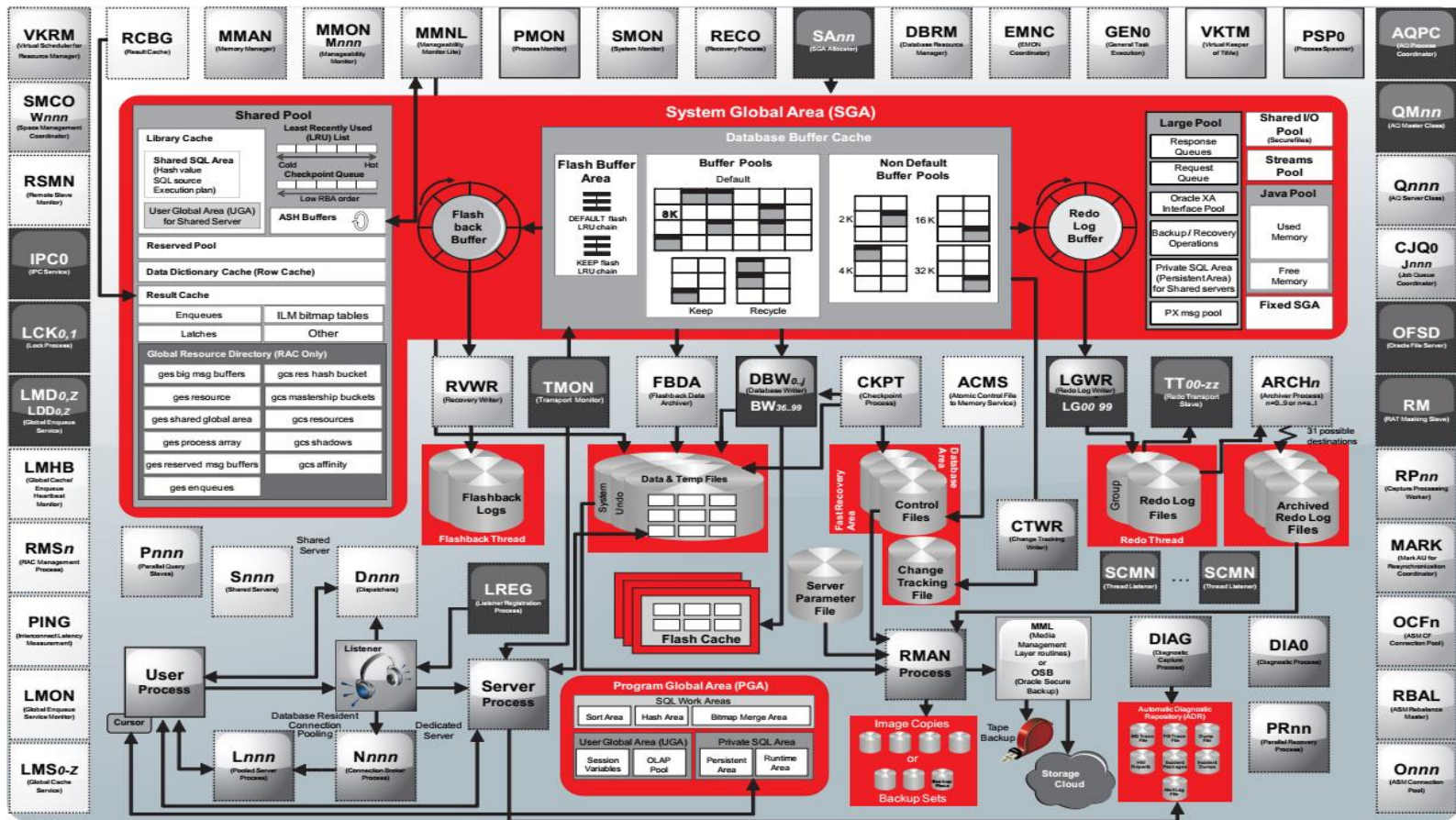


What are the background processes doing?

- Please refer to the documentation! 😊
 - Oracle 12c Database Reference, Appendix F, Background processes
- Many processes have an illustrative and precise name
 - Database Block Writer (DBWn) writes dirty blocks to disk
 - Log Writer (LGWR) flushes the contents of the Log Buffer into the redo log files.
- Other names are rather misleading
 - Checkpoint (CKPT) does not perform the checkpoints – instead, it updates the file headers at the end of each checkpoint.
 - Distributed Database Recovery process (RECO) has nothing to do with database recovery. It attempts to rescue *in-doubt transactions* that could not be completed during a *two-phase commit*.
- Again others sound mysterious:
 - Virtual Keeper of TiMe (VKTM) provides a wall clock time and reference time for time interval measurements
- I shall not explain more – but I have a nice picture!



ORACLE 12c DATABASE Architecture Diagram



Source: www.oracle.com/technetwork/tutorials/tutorials/posterfiles-1974103.pdf



Copyright © 2012, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

What are the background processes doing? – 2

- Many background processes can have several instances running in parallel
- Configuration nearly always automatically by Oracle
 - Starting with 12c, LGWR will spawn up to 100 additional workers (*LGnn*) depending on load
- DBWn can be configured to have up to 100 parallel processes
 - Parameter `DB_WRITERPROCESSES`
 - Initial setup optimized automatically
 - Very rarely necessary to adjust
 - Storage system does not support asynchronous I/O
 - Extremely heavy writing on a system with lots of CPUs



What are the background processes doing? – 3

- Every process needs CPU cycles and other resources
- Built-in Oracle service *SYS\$BACKGROUND* includes all background processes
- *V\$SERVICE_STATS*, *V\$SYS_TIME_MODEL* and others

```
SQL> desc v$service_stats
```

Name	Null?	Type
SERVICE_NAME_HASH		NUMBER
SERVICE_NAME		VARCHAR2 (64)
STAT_ID		NUMBER
STAT_NAME		VARCHAR2 (64)
VALUE		NUMBER
CON_ID		NUMBER



What are the background processes doing? – 4

```
SQL> select service_name, stat_name, value
2  from v$sqlservice_stats
3  where service_name = 'SYS$BACKGROUND'
4  order by stat_name;
```

SERVICE_NAME	STAT_NAME	VALUE
SYS\$BACKGROUND	DB CPU	26994
	DB time	6528
	application wait time	123254
	concurrency wait time	3922469
	db block changes	10844
	execute count	0

	logons cumulative	150
	opened cursors cumulative	61178

	physical reads	5539
	physical writes	813
	redo size	2528680
	session cursor cache hits	57841
	session logical reads	498338
	sql execute elapsed time	0
	user I/O wait time	215193
	user calls	2
	user commits	339
	user rollbacks	1
	workarea executions - multipass	0
	workarea executions - onepass	1
	workarea executions - optimal	9526

The *VALUE* column shows the values in microseconds or counts, depending on the metric.



What are the background processes doing? – 5

- Compare the statistics of different services

```
SQL> break on stat_name skip 1
SQL> select stat_name, service_name, value
  2  from v$sqlservice_stats
  3  order by stat_name, service_name
```

STAT_NAME	SERVICE_NAME	VALUE
DB CPU	SYS\$BACKGROUND	26994
	SYS\$USERS	1590731
DB time	SYS\$BACKGROUND	6528
	SYS\$USERS	2725734
...



Oracle12c Multithreaded Execution

- Slide 5: "In this presentation, Process equals Thread"
 - But not any more!
- Oracle default behaviour:
 - Windows: 1 process, many threads – you can still see the Oracle processes in V\$PROCESS
 - Unix/Linux: many processes – visible also on the OS
 - Including 12c
- Looking at the process listing on Unix/Linux can be intimidating...
- Minimal konfiguration, no user processes – just backgrounds
- `$ ps -eaf | grep DOAG | grep -v grep`



Oracle12c Multithreaded Execution – 2

```
oracle 11901 1 0 05:51 ? 00:00:00 ora_pmon_DOAG oracle 11977 1 0 05:51 ? 00:00:00 ora_w001_DOAG
oracle 11903 1 0 05:51 ? 00:00:00 ora_psp0_DOAG oracle 11979 1 0 05:51 ? 00:00:00 ora_aqpc_DOAG
oracle 11905 1 1 05:51 ? 00:00:01 ora_vkrm_DOAG oracle 11981 1 0 05:51 ? 00:00:00 ora_cjq0_DOAG
oracle 11909 1 0 05:51 ? 00:00:00 ora_gen0_DOAG oracle 11985 1 0 05:51 ? 00:00:00 ora_p003_DOAG
oracle 11911 1 0 05:51 ? 00:00:00 ora_mman_DOAG oracle 11987 1 0 05:51 ? 00:00:00 ora_p004_DOAG
oracle 11915 1 0 05:51 ? 00:00:00 ora_diag_DOAG oracle 11989 1 0 05:51 ? 00:00:00 ora_p005_DOAG
oracle 11917 1 0 05:51 ? 00:00:00 ora_dbrm_DOAG oracle 11991 1 0 05:51 ? 00:00:00 ora_p006_DOAG
oracle 11919 1 0 05:51 ? 00:00:00 ora_vkrm_DOAG oracle 11993 1 0 05:51 ? 00:00:00 ora_p007_DOAG
oracle 11921 1 0 05:51 ? 00:00:00 ora_dia0_DOAG oracle 11995 1 0 05:51 ? 00:00:00 ora_p008_DOAG
oracle 11923 1 0 05:51 ? 00:00:00 ora_dbw0_DOAG oracle 11997 1 0 05:51 ? 00:00:00 ora_p009_DOAG
oracle 11925 1 0 05:51 ? 00:00:00 ora_lgwr_DOAG oracle 11999 1 0 05:51 ? 00:00:00 ora_p00a_DOAG
oracle 11927 1 0 05:51 ? 00:00:00 ora_ckpt_DOAG oracle 12001 1 0 05:51 ? 00:00:00 ora_p00b_DOAG
oracle 11929 1 0 05:51 ? 00:00:00 ora_lg00_DOAG oracle 12003 1 0 05:51 ? 00:00:00 ora_p00c_DOAG
oracle 11931 1 0 05:51 ? 00:00:00 ora_smon_DOAG oracle 12005 1 0 05:51 ? 00:00:00 ora_p00d_DOAG
oracle 11933 1 0 05:51 ? 00:00:00 ora_lg01_DOAG oracle 12007 1 0 05:51 ? 00:00:00 ora_p00e_DOAG
oracle 11935 1 0 05:51 ? 00:00:00 ora_reco_DOAG oracle 12009 1 0 05:51 ? 00:00:00 ora_p00f_DOAG
oracle 11937 1 0 05:51 ? 00:00:00 ora_lreg_DOAG oracle 12157 1 0 05:51 ? 00:00:00 ora_qm02_DOAG
oracle 11939 1 0 05:51 ? 00:00:00 ora_pxmn_DOAG oracle 12159 1 0 05:51 ? 00:00:00 ora_qm03_DOAG
oracle 11941 1 0 05:51 ? 00:00:01 ora_mmon_DOAG oracle 12161 1 0 05:51 ? 00:00:00 ora_q002_DOAG
oracle 11943 1 0 05:51 ? 00:00:00 ora_mmn1_DOAG oracle 12163 1 0 05:51 ? 00:00:00 ora_q003_DOAG
oracle 11945 1 0 05:51 ? 00:00:00 ora_d000_DOAG oracle 12165 1 0 05:51 ? 00:00:00 ora_q004_DOAG
oracle 11947 1 0 05:51 ? 00:00:00 ora_s000_DOAG
oracle 11963 1 0 05:51 ? 00:00:00 ora_p000_DOAG
oracle 11965 1 0 05:51 ? 00:00:00 ora_p001_DOAG
oracle 11967 1 0 05:51 ? 00:00:00 ora_p002_DOAG
oracle 11969 1 0 05:51 ? 00:00:00 ora_tmon_DOAG
oracle 11971 1 0 05:51 ? 00:00:00 ora_tt00_DOAG
oracle 11973 1 0 05:51 ? 00:00:00 ora_smco_DOAG
oracle 11975 1 0 05:51 ? 00:00:00 ora_w000_DOAG
```

```
$ ps -eaf | grep DOAG | grep -v grep | wc -l
```

50



Oracle12c Multithreaded Execution – 3

- Just backgrounds – no sessions!
- Depending on the configuration, further processes can exist
 - At least ARCn should be there in every production database!
 - RAC, ASM, Data Guard etc. will add numerous backgrounds and slaves
 - Plus N server processes for user sessions...
- Imagine more than one database instance on the same server
 - A German customer mit 250+ instances on one Exadata
- Consider Oracle12c Multithreaded Execution
 - Most of the conventional processes, including server processes, as threads within containers
 - Less processes -> less and faster OS context switches -> possibly better performance



Oracle12c Multithreaded Execution – 4

- Turn on Multithreaded Execution and bounce the database
 - `alter system set threaded_execution = true scope=spfile;`
 - `shutdown immediate`
 - `startup`

```
$ ps -eaf | grep DOAG | grep -v grep
oracle  31781      1  0 06:49 ?          00:00:00 ora_pmon_DOAG
oracle  31783      1  0 06:49 ?          00:00:00 ora_psp0_DOAG
oracle  31785      1  1 06:49 ?          00:00:22 ora_vktm_DOAG
oracle  31789      1  0 06:49 ?          00:00:01 ora_u004_DOAG
oracle  31795      1  0 06:49 ?          00:00:13 ora_u005_DOAG
oracle  31802      1  0 06:49 ?          00:00:00 ora_dbw0_DOAG
```



Oracle12c Multithreaded Execution – 5

- Six processes instead of 50
- A few old friends...
 - DBW0
 - PMON – process monitor -> makes sense
 - PSP0 – process spawner -> ditto
 - VKTM
- ... plus new acquaintances
 - U00n
 - Container processes for the rest of the backgrounds, as well as for the server processes



Oracle12c Multithreaded Execution – 6

- V\$PROCESS still lists all individual processes
- SPID column now displays the OS container process ID
- New columns:
 - STID = OS thread ID
 - SOSID = SPID_STID
 - EXECUTION_TYPE = THREAD | PROCESS

```
SQL> select pname, sosid, spid, stid, execution_type  
       2 from v$process order by 1
```

PNAME	SOSID	SPID	STID	EXECUTION_
...				
CKPT	31789_31804	31789	31804	THREAD
...				
DBW0	31802	31802	31802	PROCESS
...				
LGWR	31789_31803	31789	31803	THREAD



Oracle12c Multithreaded Execution – 7

- Enable client connections to run in threads
 - Add the following line to LISTENER.ORA
DEDICATED_THROUGH_BROKER_LISTENER = ON
 - Restart listener

```
select s.sid, s.serial#, s.username, p.spid, p.stid, p.execution_type
from v$session s, v$process p
where s.PADDR=p.ADDR
and s.username='EMATTILA';
```

SID	SERIAL#	USERNAME	SPID	STID	EXECUTION_
13	8133	EMATTILA	31795	13969	THREAD



Oracle12c Multithreaded Execution – 8

- Kill sessions as you used to – but only from within the database!

```
alter system kill session '13,8133';  
System altered.
```

- Do **NOT** use OS command `kill -9 <pid>`
 - You will kill all sessions running in the same thread!
- OS authentication no longer possible with Multithreaded Execution
 - No OPS\$ connections (insecure, anyhow)
 - No `connect / as sysdba` (obsolete with Multitenant Option, anyhow)
 - Check your RMAN and other scripts! Also `dbstart` and `dbshut` must be adjusted – ouch.



Oracle12c Multithreaded Execution – conclusions

- Very new – not much experience, yet
- Not very well documented, yet
 - Best practises, consequences
- Potential for huge process and memory savings
 - Less PGAs due to less processes
 - UGA allocated within SGA, as with Shared Servers
 - > SGA needs more memory!
- Typical candidates are very large, consolidated systems with very many non-container databases (Exadata)



Thank you!





The power to do more