performing databases

Von lauten Nachbarn und störenden Kindern

im Wohnblock

Oracle Multitenant Performance

Martin Klier

Performing Databases GmbH
Mitterteich / Germany

Speaker

- Martin Klier
- Solution Architect and Database Expert
- My focus:
    - Performance + Tuning
    - Highly available systems
    - Cluster and Replication

- Linux since 1997
- Oracle Database since 2003

ORACLE ACE Director

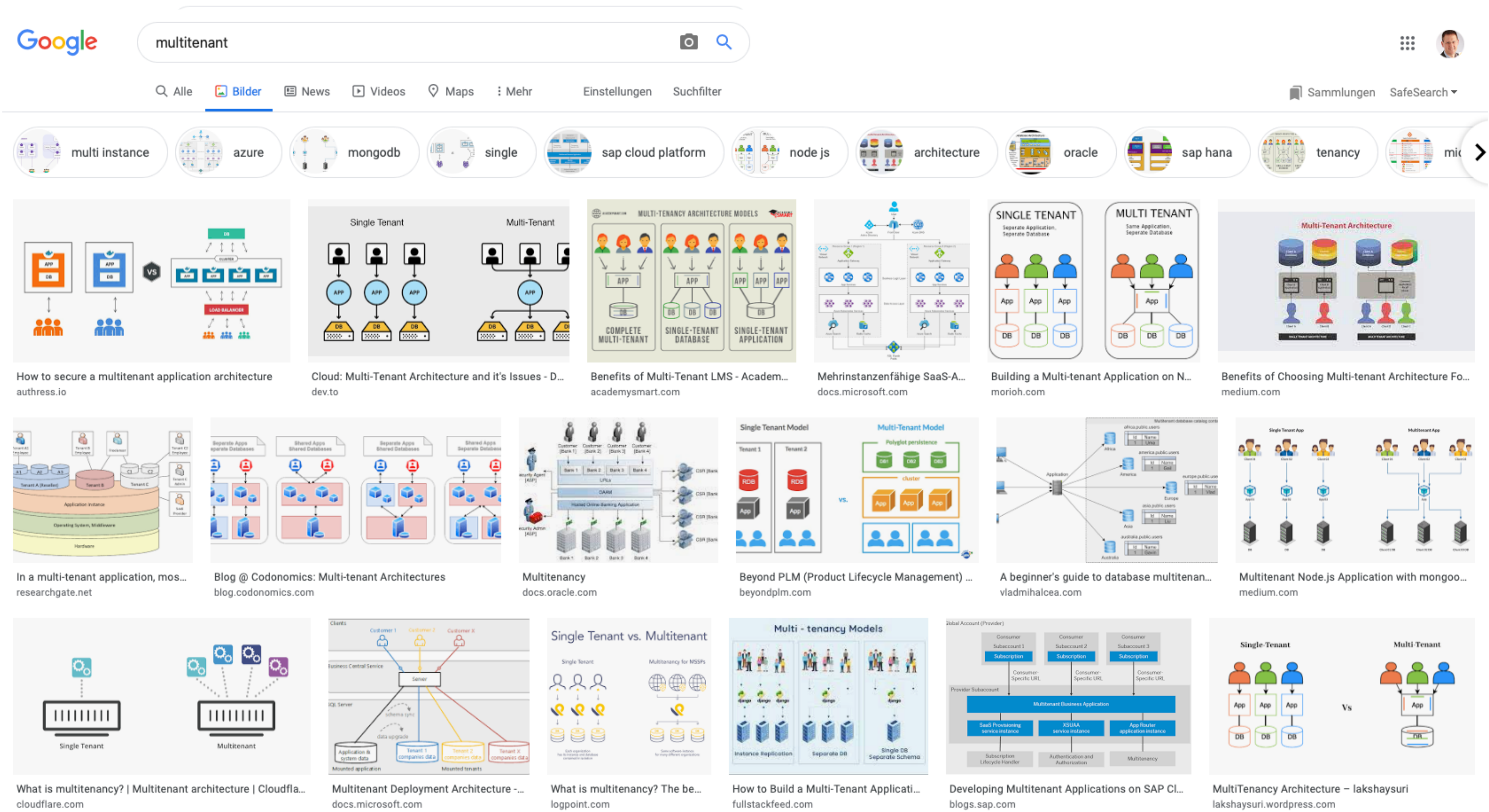
Performing Databases

- Three Experts for Database technology
  - Concepts and Project Competence
  - Architecture- and System planning
  - Licensing
  - Implementation and Troubleshooting

- Get in touch
  - Performing Databases GmbH
    Wiesauer Strasse 27
    95666 Mitterteich // Germany
  - http://www.performing-databases.com
  - Twitter: @PerformingDB

Disclaimer:

Wer sich vom „Lärm“ lebender Menschenkinder gestört fühlt, sollte unbedingt seine Prioritäten prüfen.


Google
multitenant
Alle
Bilder
News
Videos
Maps
Mehr
Einstellungen
Suchfilter
Sammlungen
SafeSearch
multi instance
azure
mongodb
single
sap cloud platform
node js
architecture
oracle
sap hana
tenancy
How to secure a multitenant application architecture
authress.io
Cloud: Multi-Tenant Architecture and it's Issues - D...
dev.to
Benefits of Multi-Tenant LMS - Academ...
academysmart.com
Mehrinstanzenfähige SaaS-A...
docs.microsoft.com
Building a Multi-tenant Application on N...
morioh.com
Benefits of Choosing Multi-tenant Architecture Fo...
medium.com
In a multi-tenant application, mos...
researchgate.net
Blog @ Codonomics: Multi-tenant Architectures
blog.codonomics.com
Multitenancy
docs.oracle.com
Beyond PLM (Product Lifecycle Management) ...
beyondplm.com
A beginner's guide to database multitenan...
vladmihalcea.com
Multitenant Node.js Application with mongoo...
medium.com
What is multitenancy? | Multitenant architecture | Cloudfla...
cloudflare.com
Multitenant Deployment Architecture -...
docs.microsoft.com
What is multitenancy? The be...
logpoint.com
How to Build a Multi-Tenant Applicati...
fullstackfeed.com
Developing Multitenant Applications on SAP Cl...
blogs.sap.com
MultiTenancy Architecture – lakshaysuri
lakshaysuri.wordpress.com

Oracle Multitenant „Pluggable Database“

Generations

I

II

III

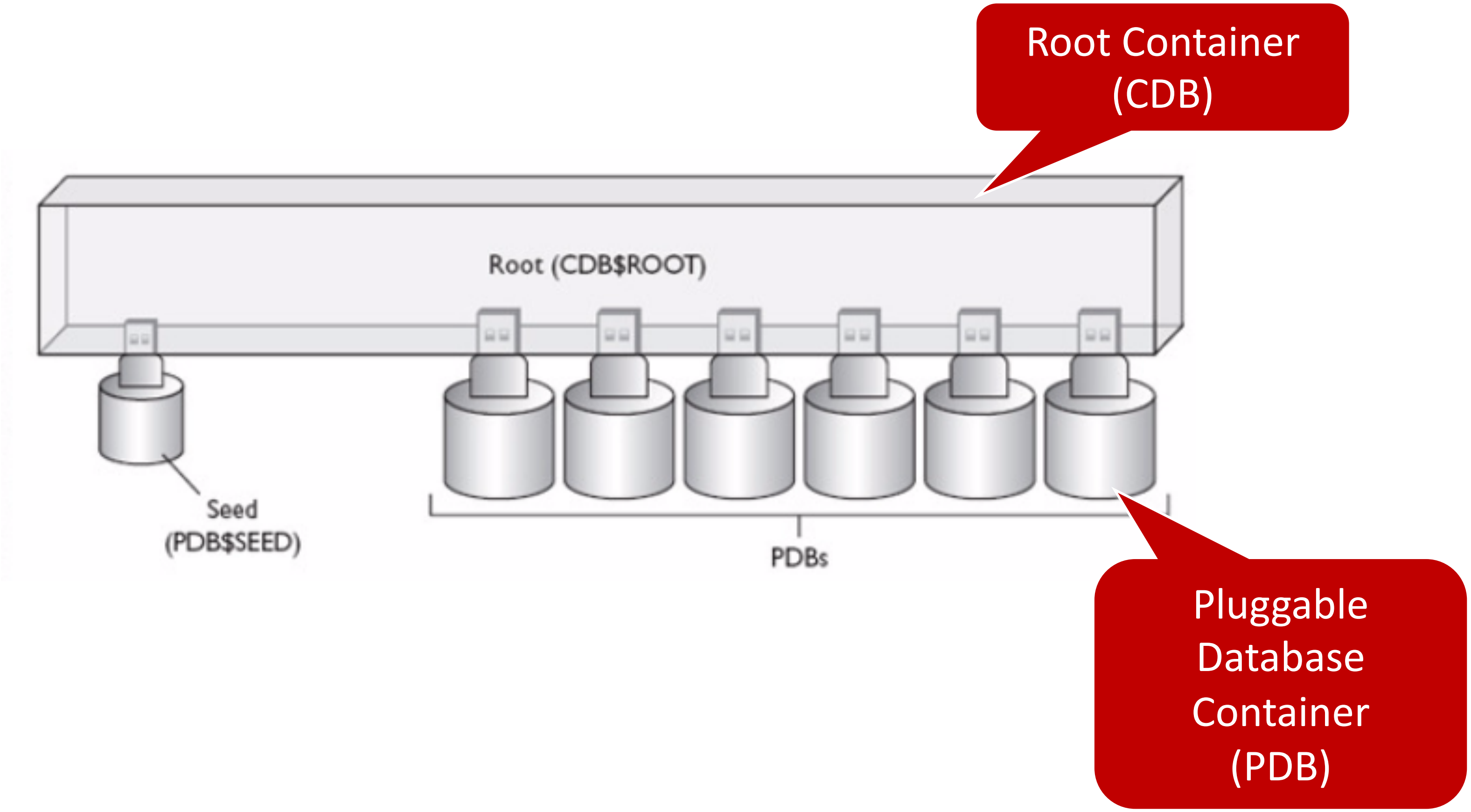
12.1

12.2
19c

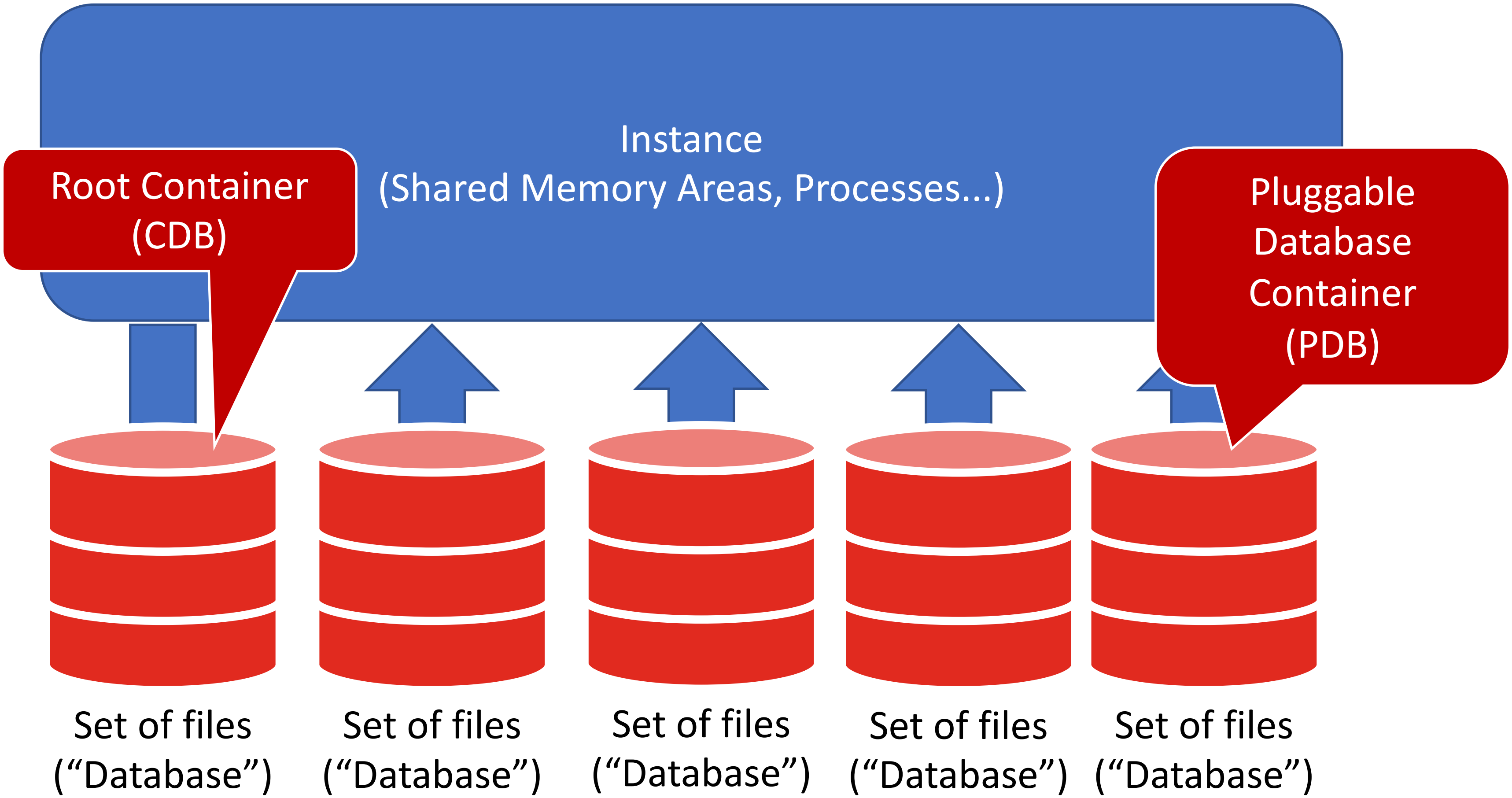
21c

<curious>

Terminology



Architecture (simplified)



Admin Challenges

performing databases

Check where you are!

```
select sys_context('Userenv','Con_name') "current container" from dual;
```

ORACLE_PDB_SID

```
[oracle@ora-bench1 ~]$ (PDB1@MT) sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Mon May 17 14:15:08 2021
Version 19.9.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.9.0.0.0

SQL> select sys_context('Userenv','Con_name') "current container" from dual;

current container
--------------------------------------------------------------------------------
CDB$ROOT
```

ORACLE_PDB_SID

```
[oracle@ora-bench1 ~]$ (PDB19MT) sqlplus sys/oracle@ora-bench1:1521/plug1 as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Mon May 17 14:27:47 2021
Version 19.9.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.9.0.0.0

SQL> select sys_context('Userenv','Con_name') "current container" from dual;

current container
--------------------------------------------------------------------------------
PLUG1
```

ORACLE_PDB_SID

```
[[oracle@ora-bench1 ~]$ (PDB19MT) export ORACLE_PDB_SID=PLUG1
[[oracle@ora-bench1 ~]$ (PDB19MT) sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Mon May 17 14:15:43 2021
Version 19.9.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.9.0.0.0

[SQL> select sys_context('Userenv','Con_name') "current container" from dual;

current container
--------------------------------------------------------------------------------
PLUG1
```

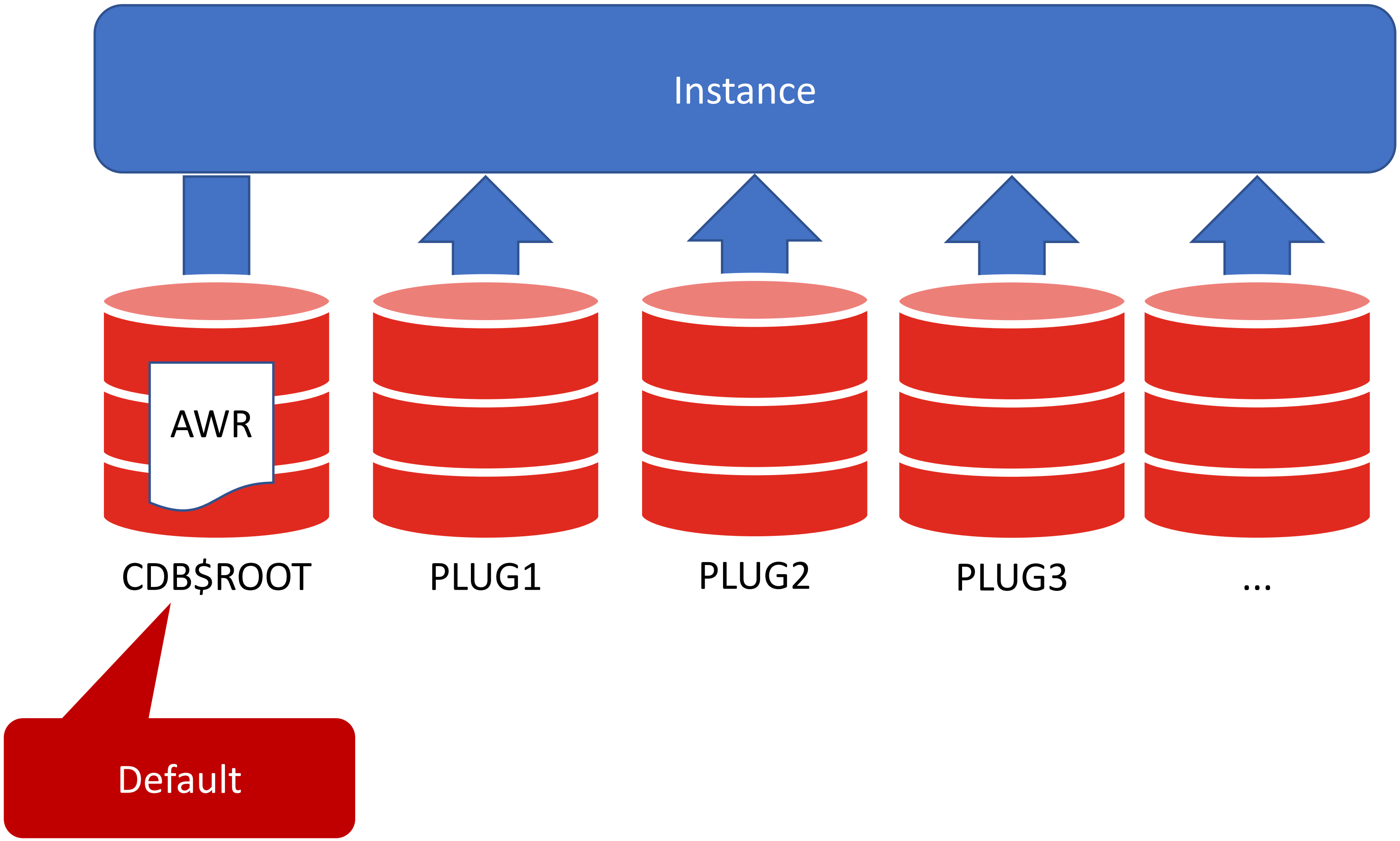
ORACLE_PDB_SID

```
[oracle@ora-bench1 ~]$ (PDB19MT) export ORACLE_PDB_SID=PLUGXXX
[oracle@ora-bench1 ~]$ (PDB19MT) sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Mon May 17 14:16:01 2021
Version 19.9.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.9.0.0.0

SQL> select sys_context('Userenv','Con_name') "current container" from dual;

current container
--------------------------------------------------------------------------------
CDB$ROOT
```

Automatic Workload Repository
AWR

performing databases

AWR



AWR

```
WORKLOAD REPOSITORY report for

DB Name      DB Id      Unique Name DB Role          Edition Release    RAC CDB
------------ ---------- ----------- ---------------- ------- ---------- --- ---
PDB19MT      4269972545 PDB19MT     PRIMARY          EE      19.0.0.0.0 NO  YES

Instance     Inst Num Startup Time    User Name    System Data Visible
------------ -------- --------------- ------------ --------------------
PDB19MT             1 17-May-21 23:16 SYS          YES

Container DB Id  Container Name       Open Time
--------------- --------------- ---------------
      212639503 PLUG3           17-May-21 23:16
      982026599 PLUG2           17-May-21 23:16
     1418868967 PDB$SEED        17-May-21 23:16
     3312395066 PLUG1           17-May-21 23:16
     4269972545 CDB$ROOT        17-May-21 23:16

Host Name        Platform                         CPUs Cores Sockets Memory(GB)
---------------- -------------------------------- ---- ----- ------- ----------
ora-bench1       Linux x86 64-bit                    8     8       1      94.31

              Snap Id      Snap Time      Sessions Curs/Sess  PDBs
            --------- ------------------- -------- --------- -----
Begin Snap:      1115 17-May-21 23:17:23        62       1.2     5
  End Snap:      1116 17-May-21 23:23:13       799       2.1     5
   Elapsed:                5.84 (mins)
   DB Time:            6,616.00 (mins)
```

AWR

```
Instance Efficiency Percentages (Target 100%)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
            Buffer Nowait %:   99.81       Redo NoWait %:   50.69
            Buffer  Hit   %:   95.29    In-memory Sort %:  100.00
            Library Hit   %:   99.33        Soft Parse %:   99.37
         Execute to Parse %:   87.37         Latch Hit %:   99.93
Parse CPU to Parse Elapsd %:    0.87     % Non-Parse CPU:   96.03
          Flash Cache Hit %:    0.00

Top 10 Foreground Events by Total Wait Time
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                                           Total Wait       Avg   % DB Wait
Event                                Waits Time (sec)      Wait   time Class
------------------------------ ----------- ---------- --------- ------ --------
resmgr:cpu quantum                  80,569     209.6K 2601.93ms   52.8 Schedule
buffer busy waits                   53,835      61.8K 1147.69ms   15.6 Concurre
enq: TX - index contention           3,117      40.6K  13.03 s    10.2 Concurre
db file sequential read            185,869      16.1K  86.74ms     4.1 User I/O
log file switch (checkpoint in         437      15.3K  34.95 s     3.8 Configur
read by other session               17,269     7639.5 442.38ms     1.9 User I/O
log file sync                       20,472     5393.5 263.46ms     1.4 Commit
db file scattered read              37,285     3709.8  99.50ms      .9 User I/O
db file parallel read               31,997     3413.7 106.69ms      .9 User I/O
library cache load lock              5,430     3276.5 603.41ms      .8 Concurre
```

??

AWR

```
Foreground Wait Events              DB/Inst: PDB19MT/PDB19MT  Snaps: 1115-1116
-> s  - second, ms - millisecond, us - microsecond, ns - nanosecond
-> Only events with Total Wait Time (s) >= .001 are shown
-> ordered by wait time desc, waits desc (idle events last)
-> %Timeouts: value of 0 indicates value was < .5%.  Value of null is truly 0

                                                     Total
                                          %Time      Wait                 Waits   % DB
Event                               Waits -outs Time (s)  Avg wait       /txn   time
-------------------------- ----------- ----- -------- --------- -------- ------
resmgr:cpu quantum              80,569         209,635 2601.93ms      3.6   52.8
buffer busy waits               53,835          61,786 1147.69ms      2.4   15.6
enq: TX - index contention       3,117          40,601   13.03 s      0.1   10.2
db file sequential read        185,869          16,123   86.74ms      8.3    4.1
log file switch (checkpoin         437          15,271   34.95 s      0.0    3.8
read by other session           17,269           7,640  442.38ms      0.8    1.9
log file sync                   20,472           5,394  263.46ms      0.9    1.4
db file scattered read          37,285           3,710   99.50ms      1.7     .9
db file parallel read           31,997           3,414  106.69ms      1.4     .9
library cache load lock          5,430           3,277  603.41ms      0.2     .8
cursor: pin S wait on X          2,508           2,521 1005.02ms      0.1     .6
enq: US - contention            24,564           1,915   77.94ms      1.1     .5
```

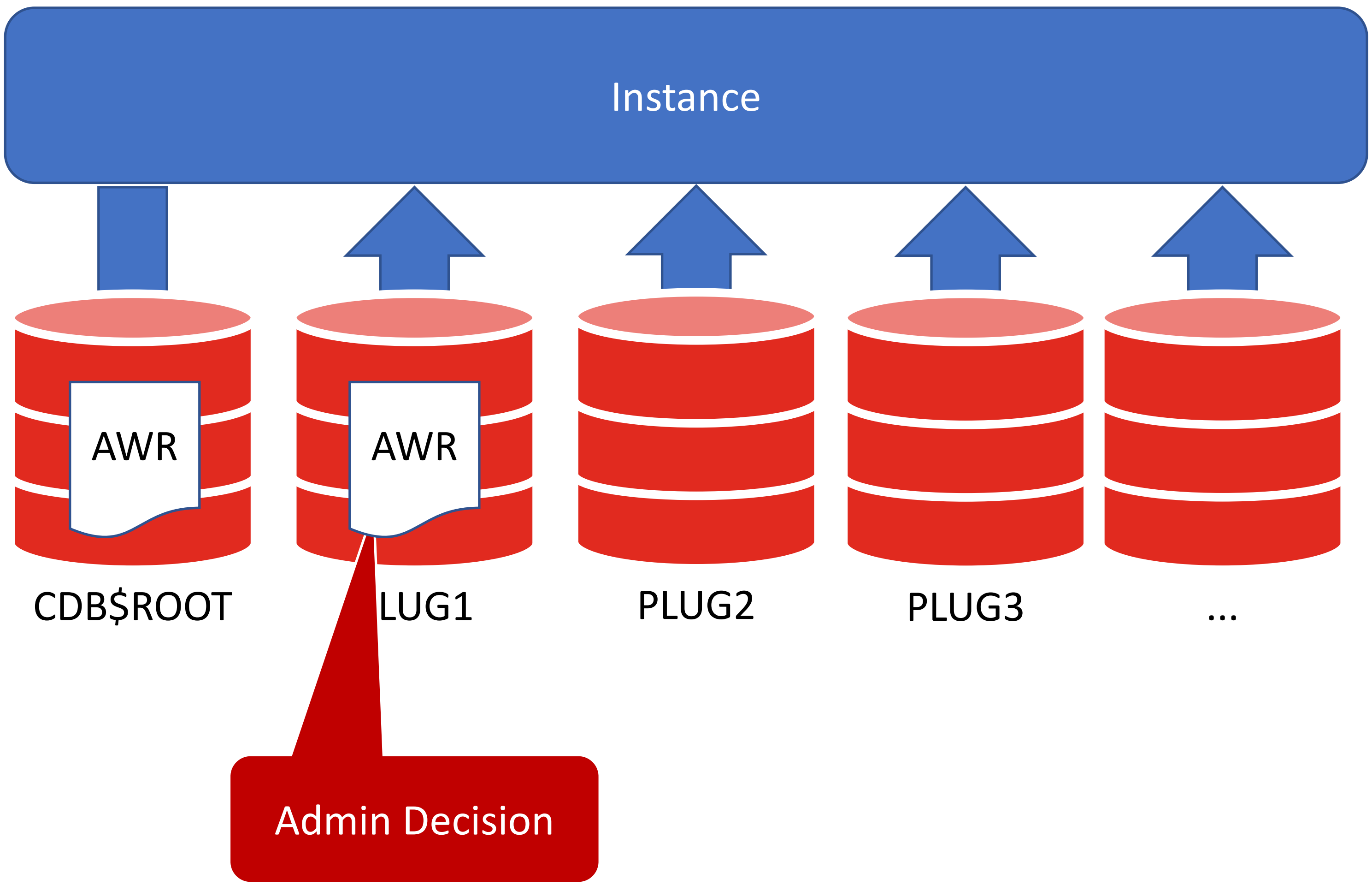
??

AWR

```
Service Statistics                 DB/Inst: PDB19MT/PDB19MT  Snaps: 1115-1116
-> ordered by DB Time

                                                           Physical      Logical
Service Name                  DB Time (s)   DB CPU (s)    Reads (K)    Reads (K)
---------------------------- ------------ ------------ ------------ ------------
plug1                             397,570          541        2,972       67,236
SYS$USERS                              81            2            2           29
PDB19MT                                 0            0            0            0
PDB19MTXDB                              0            0            0            0
SYS$BACKGROUND                          0            0            2          142
plug2                                   0            0            0            0
plug3                                   0            0            0            0
                          -------------------------------------------------------
```

```
SQL ordered by Gets                  DB/Inst: PDB19MT/PDB19MT  Snaps: 1115-1116
-> Resources reported for PL/SQL code includes the resources used by all SQL
   statements called by the code.
-> %Total - Buffer Gets   as a percentage of Total Buffer Gets
-> %CPU   - CPU Time      as a percentage of Elapsed Time
-> %IO    - User I/O Time as a percentage of Elapsed Time
-> Total Buffer Gets:      62,894,249
-> Captured SQL account for  115.7% of Total

     Buffer                 Gets              Elapsed
      Gets   Executions   per Exec   %Total  Time (s)  %CPU   %IO    SQL Id
----------- ----------- ------------ ------ ---------- ----- ----- -------------
 64,362,163         491    131,083.8  102.3   61,868.8    .5  18.1 29qp10usqkqh0
Module: Sales Rep Query
   PDB: PLUG1
SELECT TT.ORDER_TOTAL, TT.SALES_REP_ID, TT.ORDER_DATE, CUSTOMERS.CUST_FIRST_NAME
, CUSTOMERS.CUST_LAST_NAME FROM (SELECT ORDERS.ORDER_TOTAL, ORDERS.SALES_REP_ID,
 ORDERS.ORDER_DATE, ORDERS.CUSTOMER_ID, RANK() OVER (ORDER BY ORDERS.ORDER_TOTAL
 DESC) SAL_RANK FROM ORDERS WHERE ORDERS.SALES_REP_ID = :B1 ) TT, CUSTOMERS WHER

 64,353,457         488    131,871.8  102.3   61,878.4    .5  18.1 cj9v3ynkm7uuy
Module: JDBC Thin Client
   PDB: PLUG1
BEGIN :1 := orderentry.SalesRepsQuery(:2 ,:3 ,:4 ); END;

  5,078,707       9,176        553.5    8.1   93,102.9    .1  16.1 0w2qpuc6u2zsp
Module: JDBC Thin Client
   PDB: PLUG1
BEGIN :1 := orderentry.neworder(:2 ,:3 ,:4 ); END;
```

AWR



Check where you are!

```
select sys_context('Userenv','Con_name') "current container" from dual;
```

AWR in PDB

Switch to PDB – Check if you are there ;)

Manual Snapshots:

```
EXEC DBMS_WORKLOAD_REPOSITORY.create_snapshot;
```

Automatic Snapshots:

```
EXECUTE DBMS_WORKLOAD_REPOSITORY.MODIFY_SNAPSHOT_SETTINGS(
        interval => 10,    Default in PDB: 0 (=disabled)
        retention => 20160
);
```

```
[oracle@ora-bench1 ~]$ (PDB19MT) export ORACLE_PDB_SID=PLUG1
[oracle@ora-bench1 ~]$ (PDB19MT) sas
########################################################################
             Performing Databases GmbH - Mitterteich / Germany
########################################################################
This is a -#- SQLPLUS / AS SYSDBA -#- wrapper for REGULAR DATABASES
########################################################################

We are in   ORACLE_HOME = /u01/app/oracle/product/19/db_1
and we have ORACLE_SID  = PDB19MT

########################################################################

SQL*Plus: Release 19.0.0.0.0 - Production on Mon May 17 23:40:37 2021
Version 19.9.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.9.0.0.0

23:40:37 SQL PDB19MT> select sys_context('Userenv','Con_name') "current container" from dual;

current container
--------------------------------------------------------------------------------------------------
----------------------------------------------------------------------
PLUG1

Elapsed: 00:00:00.00
23:40:42 SQL PDB19MT> EXEC DBMS_WORKLOAD_REPOSITORY.create_snapshot;

PL/SQL procedure successfully completed.

Elapsed: 00:00:03.54
23:42:08 SQL PDB19MT>
23:47:25 SQL PDB19MT> EXEC DBMS_WORKLOAD_REPOSITORY.create_snapshot;

PL/SQL procedure successfully completed.

Elapsed: 00:01:54.98
23:49:24 SQL PDB19MT>
```

AWR Report in PDB

@?/rdbms/admin/awrrpt

```
Specify the location of AWR Data
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
AWR_ROOT - Use AWR data from root (default)
AWR_PDB - Use AWR data from PDB
Enter value for awr_location: AWR_PDB

Location of AWR Data Specified: AWR_PDB



Current Instance
~~~~~~~~~~~~~~~~
DB Id          DB Name        Inst Num       Instance       Container Name
-------------- -------------- -------------- -------------- --------------
 3312395066     PDB19MT                    1 PDB19MT         PLUG1


Root DB Id       Container DB Id AWR DB Id
--------------- --------------- ---------------
   4269972545       3312395066      3312395066



Instances in this Workload Repository schema
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  DB Id      Inst Num   DB Name      Instance     Host
------------ ---------- --------- ---------- ------
  3312395066      1      PDB19MT      PDB19MT      ora-bench1

Using 3312395066 for database Id
Using          1 for instance number
```

AWR Report in PDB

@?/rdbms/admin/awrrpt

```
Instances in this Workload Repository schema
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  DB Id      Inst Num  DB Name      Instance     Host
------------ ---------- ---------    ----------   ------
  3312395066     1     PDB19MT      PDB19MT      ora-bench1

Using 3312395066 for database Id
Using          1 for instance number


Specify the number of days of snapshots to choose from
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Entering the number of days (n) will result in the most recent
(n) days of snapshots being listed.  Pressing <return> without
specifying a number lists all completed snapshots.


Enter value for num_days: 1

Listing the last day's Completed Snapshots
Instance     DB Name      Snap Id       Snap Started    Snap Level
------------ ------------ ---------- ------------------ ----------

PDB19MT      PDB19MT              1  17 May 2021 23:42    1
                                  2  17 May 2021 23:47    1


Specify the Begin and End Snapshot Ids
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Enter value for begin_snap: 1
Begin Snapshot Id specified: 1

Enter value for end_snap: 2
End   Snapshot Id specified: 2
```

AWR Report in PDB

```
WORKLOAD REPOSITORY PDB report (PDB snapshots)

DB Name         DB Id    Unique Name DB Role          Edition Release    RAC CDB
------------ ----------- ----------- ---------------- ------- ---------- --- ---
PDB19MT       3312395066 PDB19MT     PRIMARY          EE      19.0.0.0.0 NO  YES

Instance     Inst Num Startup Time    User Name    System Data Visible
------------ -------- --------------- ------------ --------------------
PDB19MT             1 17-May-21 23:16 SYS          YES

Container DB Id  Container Name       Open Time
--------------- --------------- ---------------
     3312395066 PLUG1           17-May-21 23:16

Host Name        Platform                         CPUs Cores Sockets Memory(GB)
---------------- -------------------------------- ---- ----- ------- ----------
ora-bench1       Linux x86 64-bit                    8     8       1      94.31

              Snap Id      Snap Time      Sessions Curs/Sess
            --------- ------------------- -------- ---------
Begin Snap:         1 17-May-21 23:42:04         0       3.0
  End Snap:         2 17-May-21 23:47:47       475       2.4
   Elapsed:                5.70 (mins)
   DB Time:            6,760.25 (mins)
```

AWR Report in PDB

```
Load Profile                    Per Second   Per Transaction  Per Exec  Per Call
~~~~~~~~~~~~~~~            ---------------   --------------- --------- ---------
             DB Time(s):           1,185.0              11.4      0.40      1.97
              DB CPU(s):               3.2               0.0      0.00      0.01
      Background CPU(s):               0.0               0.0      0.00      0.00
      Redo size (bytes):         422,776.7           4,068.4
  Logical read (blocks):         293,080.3           2,820.3
          Block changes:           2,818.4              27.1
 Physical read (blocks):           3,785.3              36.4
Physical write (blocks):               0.3               0.0
       Read IO requests:             708.5               6.8
      Write IO requests:               0.1               0.0
           Read IO (MB):              29.6               0.3
          Write IO (MB):               0.0               0.0
           IM scan rows:               0.0               0.0
Session Logical Read IM:               0.0               0.0
             User calls:             600.8               5.8
           Parses (SQL):             303.9               2.9
      Hard parses (SQL):               0.6               0.0
     SQL Work Area (MB):               1.7               0.0
                 Logons:               2.9               0.0
            User logons:               2.9               0.0
         Executes (SQL):           2,972.3              28.6
              Rollbacks:               0.6               0.0
           Transactions:             103.9

Top 10 Foreground Events by Total Wait Time
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                                           Total Wait       Avg   % DB Wait
Event                                Waits Time (sec)      Wait   time Class
------------------------------ ----------- ---------- --------- ------ --------
resmgr:cpu quantum                  49,697       255K 5130.74ms   62.9 Schedule
db file sequential read            149,753      55.8K  372.33ms   13.7 User I/O
db file scattered read              72,122      33.9K  470.29ms    8.4 User I/O
log file sync                       36,318      31.1K  854.97ms    7.7 Commit
buffer busy waits                    5,023      14.1K 2816.99ms    3.5 Concurre
```

No
Instance Efficiency Percentages

Scalability

Scaling Multitenant

Element	Terms & Conditions
Instance: SGA	Structure shared, QoS possible
Instance: Library Cache	Structure shared, content isolated
Instance: Buffer Cache	Structure shared, content isolated
Instance: Processes	Shared
Root Container: Online Redo Facility	Shared, but “labelled” per PDB
Root Container: Control File	Shared
PDB Container: Data Files	Independent

Configure and Rule

For better isolation and individualization, you can set parameters on PDB level:

DB_CACHE_SIZE:	The minimum buffer cache size for the PDB.
SHARED_POOL_SIZE:	The minimum shared pool size for the PDB.
PGA_AGGREGATE_LIMIT:	The maximum PGA size for the PDB.
PGA_AGGREGATE_TARGET:	The target PGA size for the PDB.
SGA_MIN_SIZE:	The minimum SGA size for the PDB.
SGA_TARGET:	The maximum SGA size for the PDB.

Thanks, Tim! <3

https://oracle-base.com/articles/12c/multitenant-memory-resource-management-for-pdbs-12cr2

Library Cache

Library Cache

Very simple test case.

Running 1 different and 1 identical query in various containers.

```
#!/bin/bash
. ~/bin/db-mt >/dev/null
export ORACLE_PDB_SID=$1
sqlplus -S / as sysdba <<EOQ
select sys_context('Userenv','Con_name') "current container" from dual;
select /* LOADER_$ORACLE_PDB_SID */ count(*) from all_objects;
select /* LOADER_STATIC */ sum(object_id) from all_objects;
EOQ
```

🡐 PLUG1, PLUG2 ...

Library Cache

Query in CDB$ROOT:

```sql
select con_id, sql_id, fetches, sql_fulltext
    from v$sql
    where sql_fulltext like 'select /* LOADER%'
    order by con_id;
```

Abfrageergebnis 3

SQL | Alle Zeilen abgerufen:4 in 0,022 Sekunden

	CON_ID	SQL_ID	FETCHES	SQL_FULLTEXT
1	Plug1 3	6vf9y3t5cnbtt	10	select /* LOADER_STATIC */ sum(object_id) from all_objects
2	Plug1 3	7kn8j0hyzav2k	10	select /* LOADER_PLUG1 */ count(*) from all_objects
3	Plug2 4	6vf9y3t5cnbtt	3	select /* LOADER_STATIC */ sum(object_id) from all_objects
4	Plug2 4	59rshrbuhtfpv	3	select /* LOADER_PLUG2 */ count(*) from all_objects

Same SQL Text → Same SQL_ID in LC

→ Shared Parent Cursor

→ One Child Cursor per PDB

Library Cache

```
select con_id, username, sid, sql_id, sql_child_number, event
    from v$session
    where username='KLM'
        and SQL_ID='88x71r7fju58c'
    ;
```

Abfrageergebnis 32

SQL Alle Zeilen abgerufen:3 in 0,027 Sekunden

	CON_ID	USERNAME	SID	SQL_ID	SQL_CHILD_NUMBER	EVENT
1	Plug1 3	KLM	11	88x71r7fju58c	2	cursor: pin S
2	Plug1 3	KLM	388	88x71r7fju58c	2	cursor: pin S
3	Plug1 3	KLM	389	88x71r7fju58c	2	cursor: pin S

Hammering on the same cursor leads to “cursor: pin S” waits (expected).

Library Cache

```sql
select con_id, username, sid, sql_id, sql_child_number, event
    from v$session
    where username='KLM'
        and SQL_ID='88x71r7fju58c'
    ;
```

Abfrageergebnis 17

SQL Alle Zeilen abgerufen:3 in 0,021 Sekunden

	CON_ID	USERNAME	SID	SQL_ID	SQL_CHILD_NUMBER	EVENT
1	Plug2 4	KLM	11	88x71r7fju58c	3	PGA memory operation
2	Plug3 5	KLM	388	88x71r7fju58c	4	PGA memory operation
3	Plug1 3	KLM	389	88x71r7fju58c	2	PGA memory operation

Hammering on the same cursor <u>from different PDBs</u> causes NO CONCURRENCY (exact same workload)

➔ LC Content is well isolated among PDBs

Library Cache

```sql
select con_id, sql_id, executions, sql_fulltext
    from v$sql
    where sql_fulltext like 'select /* LOOP%'
        and rownum <100;
    --order by con_id;
```

Abfrageergebnis | Abfrageergebnis 1 | Abfrageergebnis 2 | Abfrageergebnis 3

SQL | 50 Zeilen abgerufen in 0,015 Sekunden

		CON_ID	SQL_ID	EXECUTIONS	SQL_FULLTEXT
1	Plug2	4	d01kx0zu2n001	1	select /* LOOP_22012241652665201 */ table_name from KLM.MYTABLE where ro...
2		4	9b4fx5t2cc001	1	select /* LOOP_49493920102045202 */ table_name from KLM.MYTABLE where ro...
3		4	7rn77vb0ts002	1	select /* LOOP_76062926114041399 */ table_name from KLM.MYTABLE where ro...
4		4	35vn9btpq0002	1	select /* LOOP_91747564755167414 */ table_name from KLM.MYTABLE where ro...
5	Plug1	3	f6magv3rrs003	1	select /* LOOP_48388484427382900 */ table_name from KLM.MYTABLE where ro...
6		3	26bqzadqdh004	1	select /* LOOP_87021437268837732 */ table_name from KLM.MYTABLE where ro...
7		3	1wxpz86rrh005	1	select /* LOOP_67768614750244011 */ table_name from KLM.MYTABLE where ro...
8		3	323dnmz0b8006	1	select /* LOOP_61504534119188380 */ table_name from KLM.MYTABLE where ro...
9		3	0dzq9rv4j4006	1	select /* LOOP_73064930237444428 */ table_name from KLM.MYTABLE where ro...
10		3	961vz9fgbn006	1	select /* LOOP_19332811624749551 */ table_name from KLM.MYTABLE where ro...

Full-speed flooding the LC with different SQL text leads to .....

Library Cache

```
select con_id, username, sid, sql_id, sql_child_number, event
    from v$session
    where username='KLM'
    ;
```

Abfrageergebnis | Abfrageergebnis 1

SQL | Alle Zeilen abgerufen:2 in 0,044 Sekunden

	CON_ID	USERNAME	SID	SQL_ID	SQL_CHILD_NUMBER	EVENT
1	Plug2 4	KLM	911	44a7mdzfu0d13	0	latch: shared pool
2	Plug1 3	KLM	1967	bgrxy7rbd7h5d	0	latch: shared pool

Full-speed flooding the LC with different SQL text leads to .....
..... severe latching on the shared pool.

➔ Structure is shared between PDBs

Library Cache

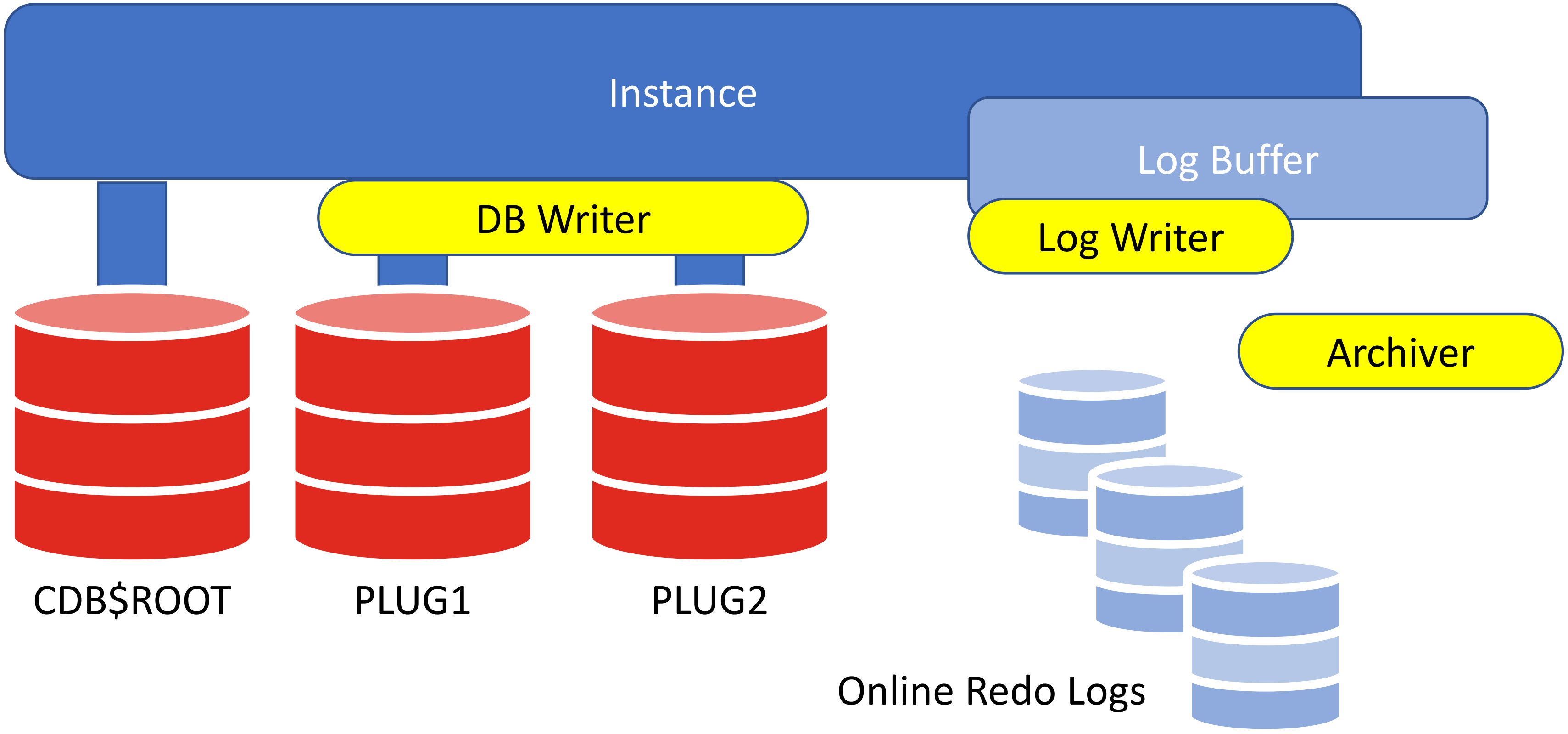
Conclusion:

- Regular use (Cursor Sharing) scales well
- Abuse (Library Cache Stress) does not

➔ Do code well, and it will scale
➔ Do code badly, and you will pay ☺

Shared Resources

Sharing Resources



Future – 21c!

Meet & Greet

martin.klier@performing-db.com

www.performing-databases.com

Many national // international events

Efficiency is
to stay in touch!

martin.klier@performing-db.com

http://www.performing-databases.com

performing
databases

performing
databases