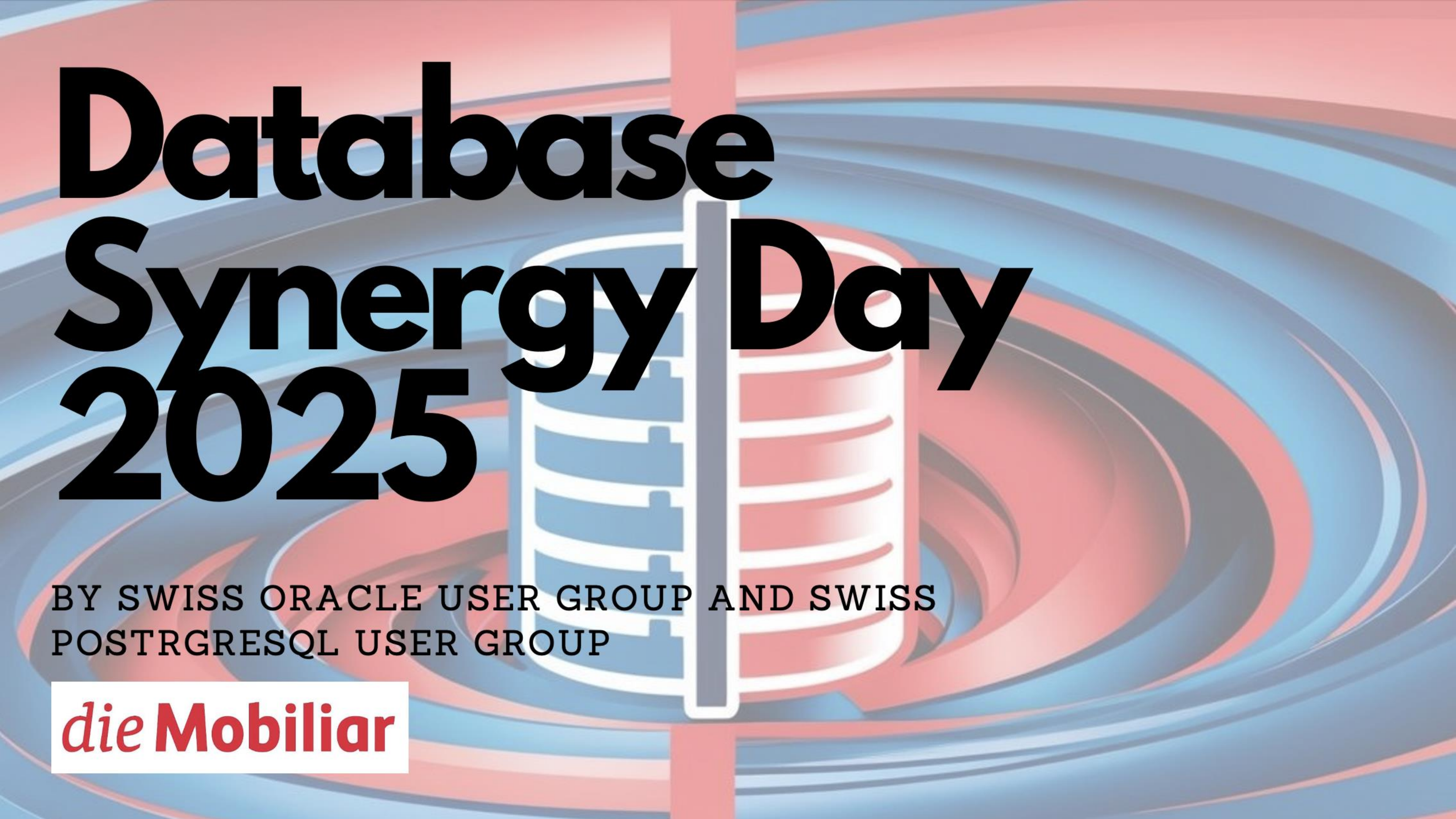
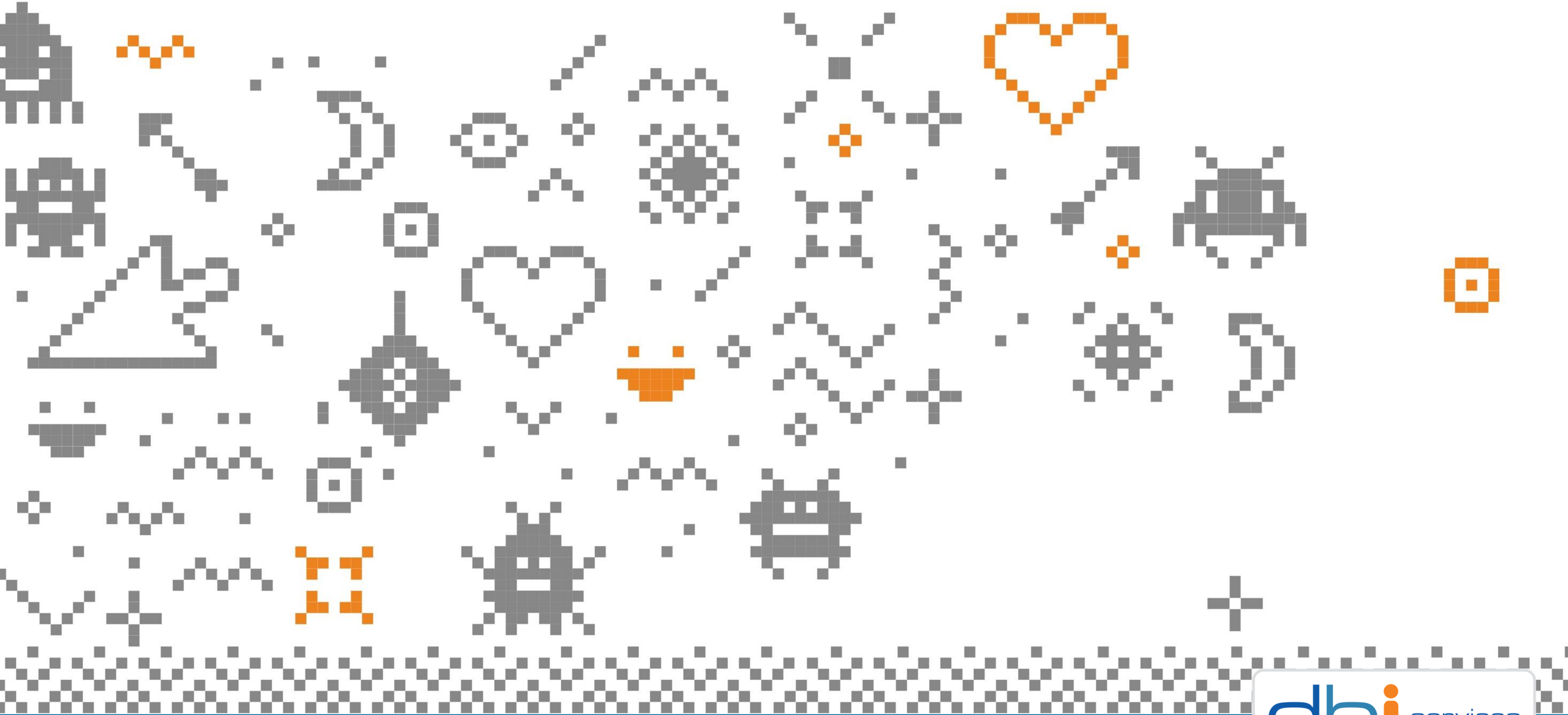


Database Synergy Day 2025

BY SWISS ORACLE USER GROUP AND SWISS POSTRGRESQL USER GROUP

die Mobiliar



Resolving Oracle Issues

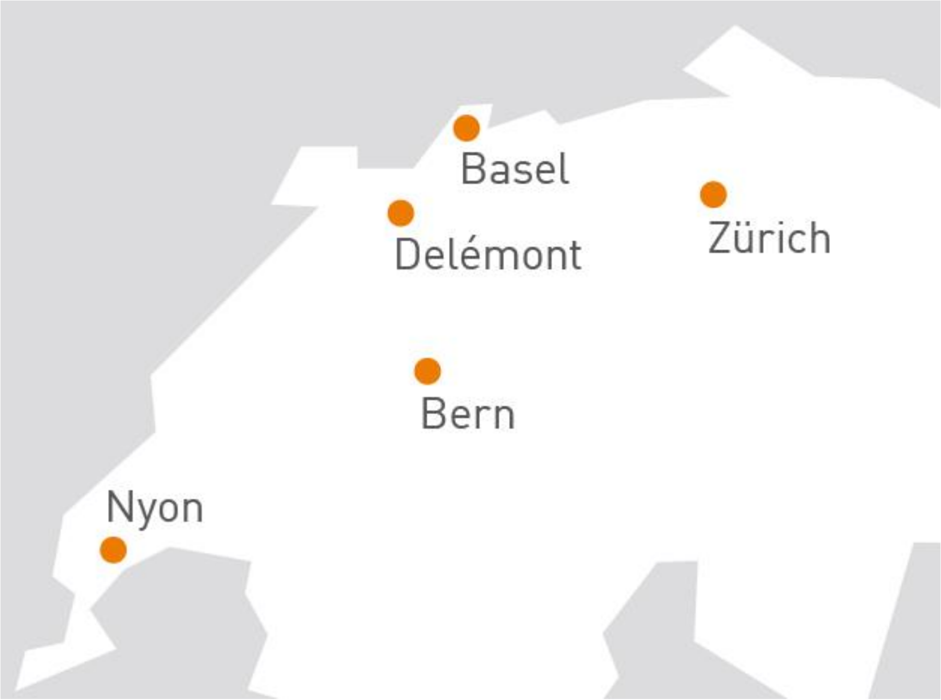
Who we are

The Company

- > Founded in 2010
- > More than 100 employees
- > Specialized in the Middleware Infrastructure
  - > The invisible part of IT
- > Customers in Switzerland and all over Europe

Our Offer

- > Consulting
- > Service Level Agreements (SLA)
- > Trainings
- > License Management



swiss safety center ISO 20000 certified system

swiss safety center ISO 27001 certified system

About me

Clemens Bleile

Principal Consultant & Technology Leader Oracle

+41 78 677 51 09

clemens.bleile[at]dbi-services.com

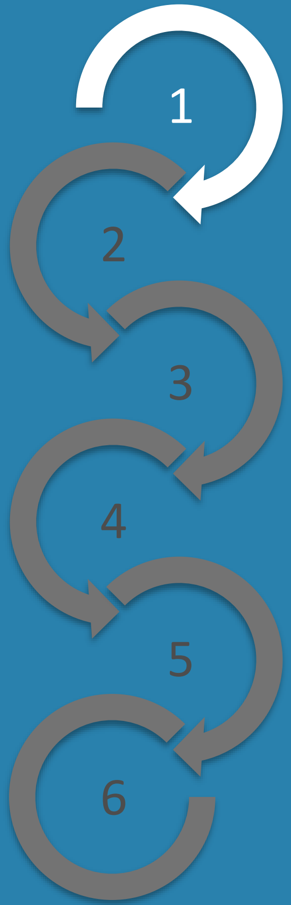
Agenda

1. Motivation
2. Approach
3. Case 1
4. Case 2
5. Tools available
6. Recommendation and Summary

ORACLE Database 23ai

Motivation

- What happened the last couple of years?



Motivation
How did you resolve that?

After I could identify the root cause of an issue and resolve it people often ask

How did you do that?

Motivation
My background in problem solving

- 2.5 years as DBA and System Admin
  - Oracle 6.0.36 and 7.0.13 with Oracle Parallel Server on OpenVMS
- 6 years in Oracle Support
  - Telephone support
- 7 years in Oracle Consulting
  - Biggest General Ledger system of the world
- 7 years in Oracle Support (Team Lead of the Performance Team)
  - Optimizer issues
  - Wrong results
  - Sev 1 issues
- Over 9 years as a Consultant and Technology Lead @ dbi services
  - Engineered Systems

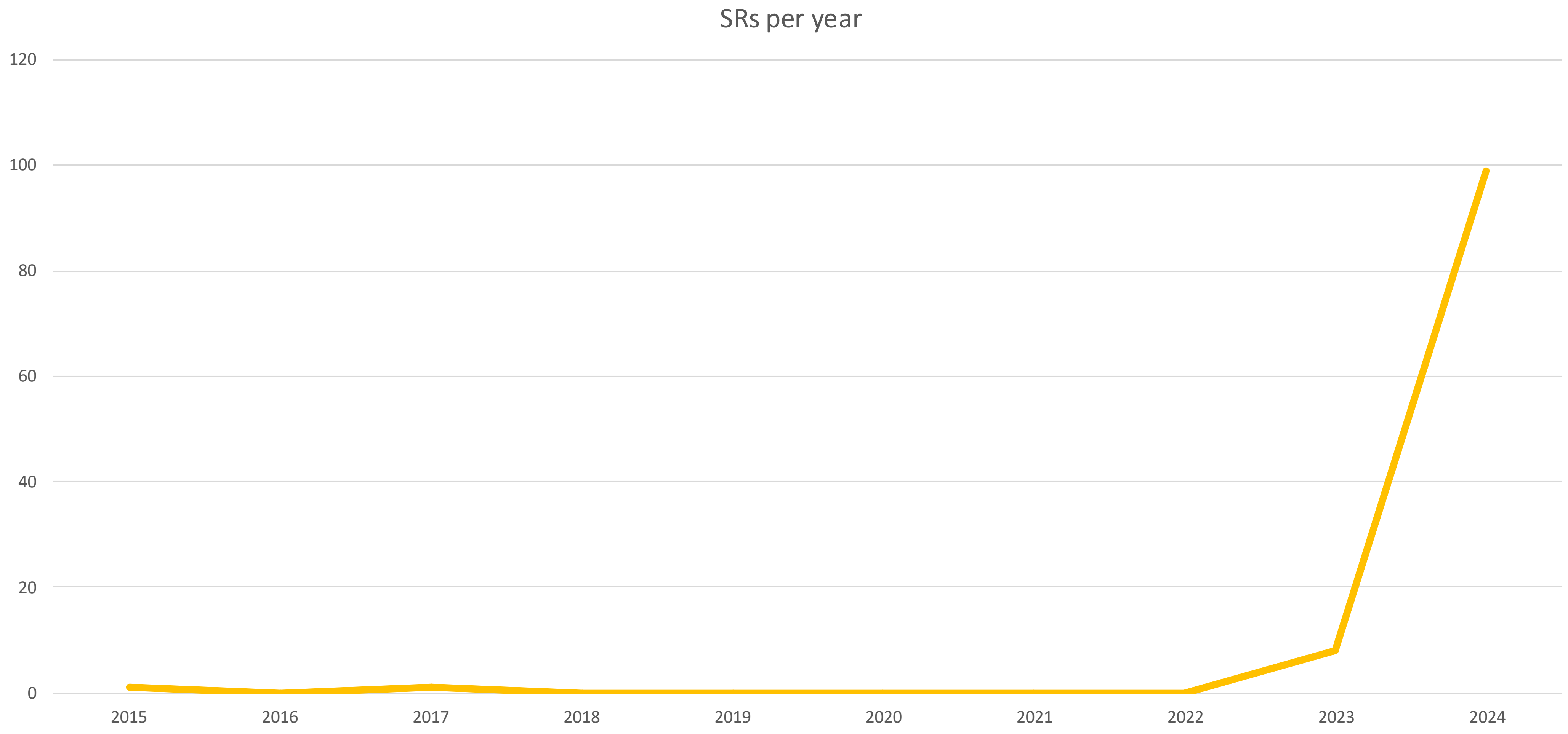
Motivation
What happened the last 25 years?

- DBs became a commodity
- More features in the DB caused it to work without much intervention
- Less people had to manage more DBs and different DBs (brand and type)
  - Broader skills required

- Due to the market situation and competition, customers tried to reduce IT costs by outsourcing, offshoring or nearshoring

- However, in case of an issue, problem resolution time usually increased
- In addition, the DB became also more complex inside its kernel due to many new features

Motivation
Customer: SR statistics



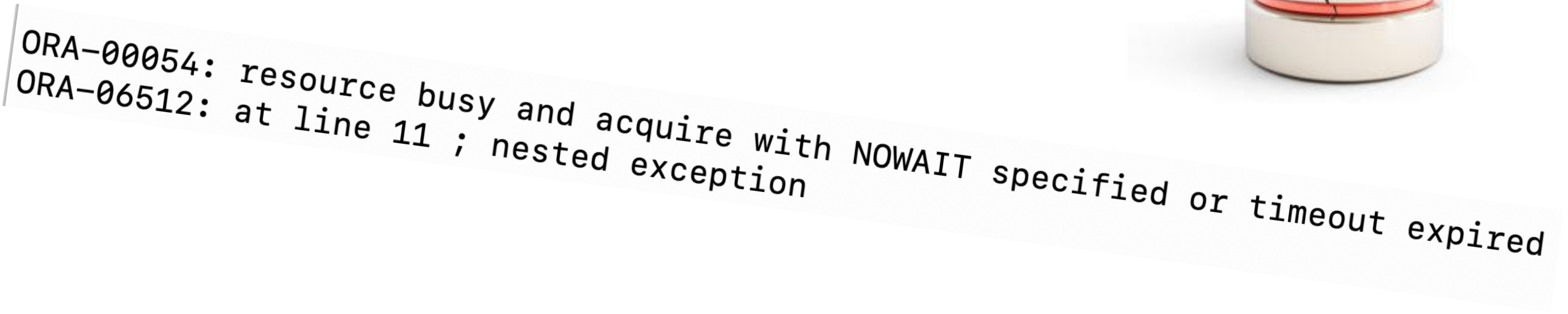
Motivation
Strange issues

- ORA-00054: resource busy and acquire with NOWAIT specified or timeout expired
- ExaCC: ASM does not start during reboot
- Performance issues
- Timeouts accessing single-node DB-clusters over the network
- Default parallelism behavior
- Timeouts connecting through CMAN
- …

```
ORA-00054: resource busy and acquire with NOWAIT specified or timeout expired
ORA-06512: at line 11 ; nested exception
```

Motivation
Strange issues

- Service provider created an Oracle Service Request (SR) for every issue
- Time consuming resolution process
  - Lots of data to provide to Oracle
  - Misunderstandings
  - Days between updates
- Customer not satisfied



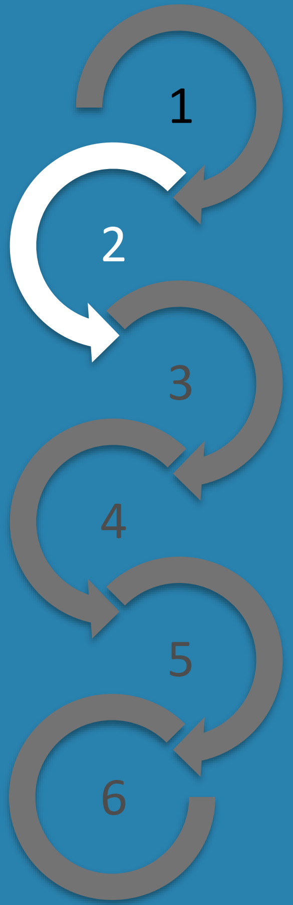
```
ORA-00054: resource busy and acquire with NOWAIT specified or timeout expired
ORA-06512: at line 11 ; nested exception
```

Motivation
How to resolve Oracle issues?

> How to resolve Oracle issues???

Approach

- Problem resolution process
- Where to find the needed data or how to get more info?



Approach
Problem resolution process

Die Autodoktoren - offizieller Kanal

@Autodoktoren · 819K subscribers · 829 videos

https://www.youtube.com/@Autodoktoren

Approach
Problem resolution process

https://www.youtube.com/shorts/aH9Y5mWc-ys

Approach
Problem resolution process

- understand the issue
- root cause analysis
- trying to understand how something works
  - in theory
  - in practice
  - by creating a model
  - by disassembling stuff
- experience helps a lot
- have good tools available
  - to disassemble and assemble
  - to monitor something
  - to readout error from the system
  - to test something related to the device I'm working on (e.g. press smoke in the engine to see leaks)

Approach
Problem resolution process

- What is the problem?
- → Problem description

- Where and how to get more information?
- → Analysis
- Use the resources available (alert.log, trace-files, MOS-search, etc.)
- Document your steps

- What is the root cause?
- → Outcome of Analysis

- How can I change the approach/process/system to resolve the issue permanently?
- → Solution

Approach
Problem resolution process

> What is the problem?

> → Problem description

I do a lot of data exports for clients with
parallel mode. I started with 32 parallel and for 1 month I can't use all parallel.

For example, I started with 32 and then I had a problem
but I can use 16 after another problem but works with 8 but now I'm at 4...

Each time I have the following problem:

Processing object type TABLE_EXPORT/TABLE/TABLE_DATA
ORA-31687: error creating worker process with worker id 5 (here the number of the worker id changes. Before it was 24 then 14 then 7 ...)
ORA-31687: error creating worker process with worker id 5
ORA-31612: The allocation of the process descriptor has failed.

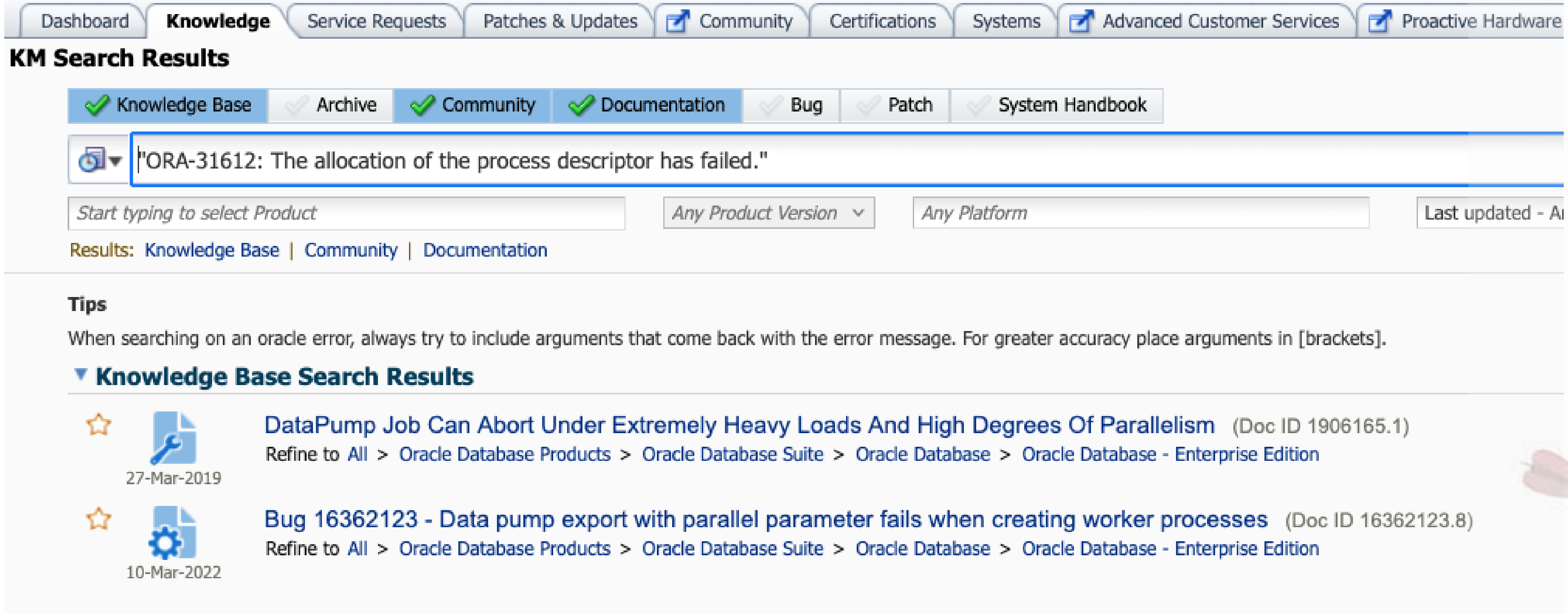
I searched Oracle support, but I couldn't find any real solutions and nothing special in alertlog. The database version is 11.2.0.3, so no more support...

Do you have any idea ?

Approach
Problem resolution process

> Where and how to get more information?

> → Analysis



KM Search Results

✓ Knowledge Base | Archive | ✓ Community | ✓ Documentation | Bug | Patch | System Handbook

"ORA-31612: The allocation of the process descriptor has failed."

Start typing to select Product | *Any Product Version* | *Any Platform* | Last updated - A

Results: Knowledge Base | Community | Documentation

Tips

When searching on an oracle error, always try to include arguments that come back with the error message. For greater accuracy place arguments in [brackets].

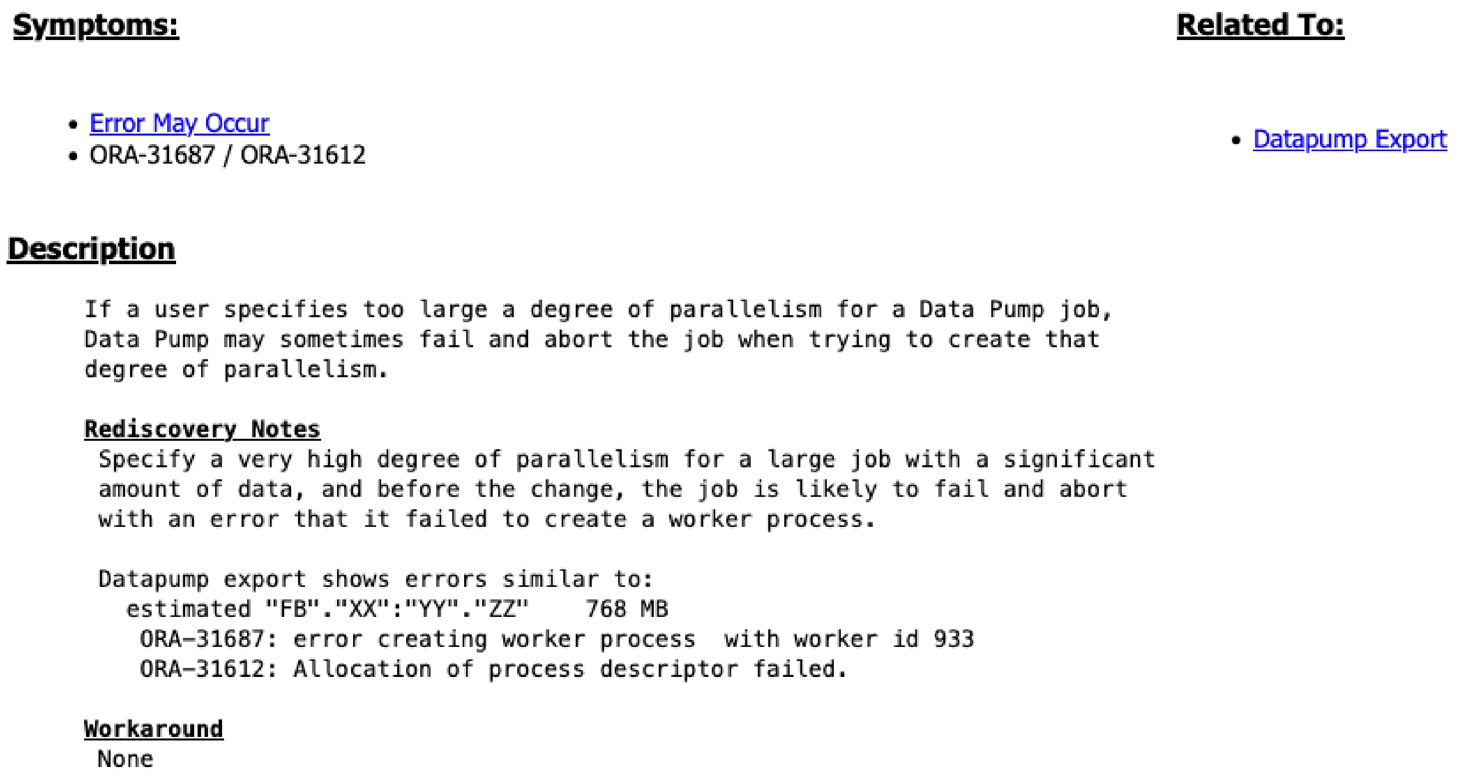
Knowledge Base Search Results

DataPump Job Can Abort Under Extremely Heavy Loads And High Degrees Of Parallelism (Doc ID 1906165.1)
Refine to All > Oracle Database Products > Oracle Database Suite > Oracle Database > Oracle Database - Enterprise Edition
27-Mar-2019

Bug 16362123 - Data pump export with parallel parameter fails when creating worker processes (Doc ID 16362123.8)
Refine to All > Oracle Database Products > Oracle Database Suite > Oracle Database > Oracle Database - Enterprise Edition
10-Mar-2022

Approach
Problem resolution process

- Where and how to get more information?
- → Analysis



Symptoms:

- Error May Occur
- ORA-31687 / ORA-31612

Related To:

- Datapump Export

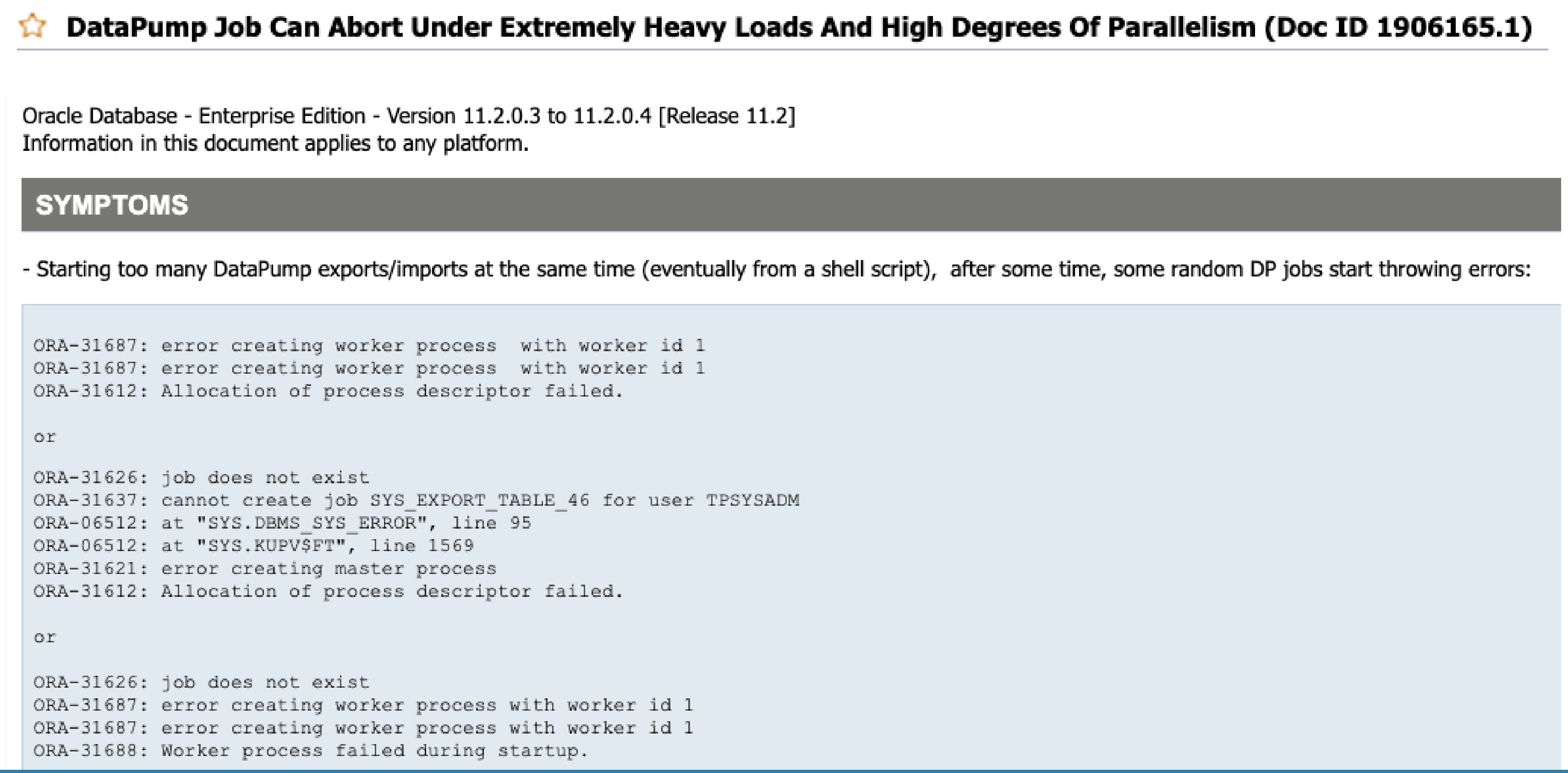
Description

```
If a user specifies too large a degree of parallelism for a Data Pump job,
Data Pump may sometimes fail and abort the job when trying to create that
degree of parallelism.

Rediscovery Notes
 Specify a very high degree of parallelism for a large job with a significant
 amount of data, and before the change, the job is likely to fail and abort
 with an error that it failed to create a worker process.

 Datapump export shows errors similar to:
   estimated "FB"."XX":"YY"."ZZ"    768 MB
    ORA-31687: error creating worker process  with worker id 933
    ORA-31612: Allocation of process descriptor failed.

Workaround
 None
```

Approach
Problem resolution process

> Where and how to get more information?

> → Analysis



☆ **DataPump Job Can Abort Under Extremely Heavy Loads And High Degrees Of Parallelism (Doc ID 1906165.1)**

Oracle Database - Enterprise Edition - Version 11.2.0.3 to 11.2.0.4 [Release 11.2]
Information in this document applies to any platform.

SYMPTOMS

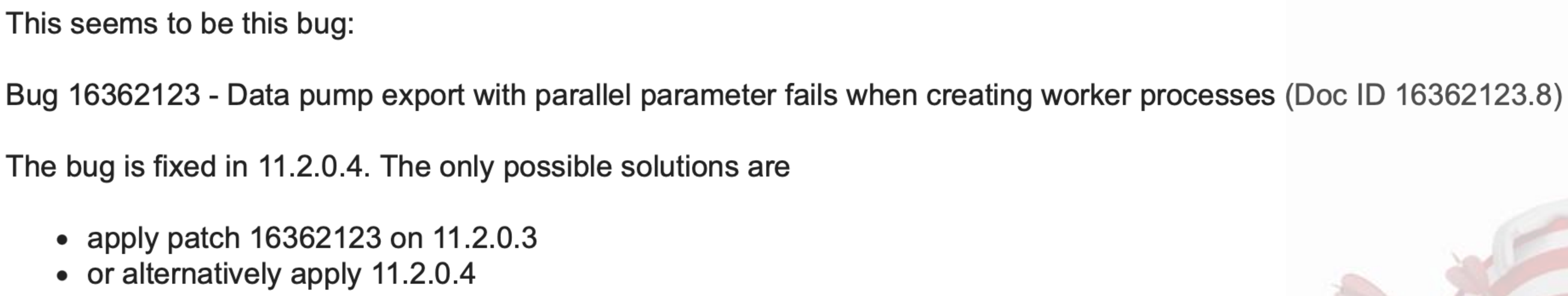
- Starting too many DataPump exports/imports at the same time (eventually from a shell script), after some time, some random DP jobs start throwing errors:

```
ORA-31687: error creating worker process  with worker id 1
ORA-31687: error creating worker process  with worker id 1
ORA-31612: Allocation of process descriptor failed.

or

ORA-31626: job does not exist
ORA-31637: cannot create job SYS_EXPORT_TABLE_46 for user TPSYSADM
ORA-06512: at "SYS.DBMS_SYS_ERROR", line 95
ORA-06512: at "SYS.KUPV$FT", line 1569
ORA-31621: error creating master process
ORA-31612: Allocation of process descriptor failed.

or

ORA-31626: job does not exist
ORA-31687: error creating worker process with worker id 1
ORA-31687: error creating worker process with worker id 1
ORA-31688: Worker process failed during startup.
```

Approach
Problem resolution process

- What is the root cause?
- → Outcome of Analysis

- How can I change the approach/process/system to resolve the issue permanently?
- → Solution



This seems to be this bug:

Bug 16362123 - Data pump export with parallel parameter fails when creating worker processes (Doc ID 16362123.8)

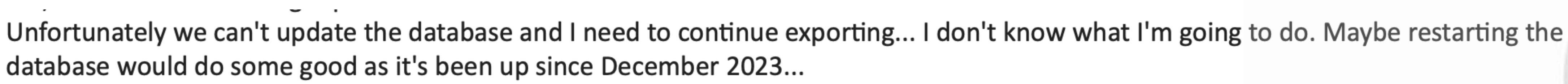
The bug is fixed in 11.2.0.4. The only possible solutions are

- apply patch 16362123 on 11.2.0.3
- or alternatively apply 11.2.0.4

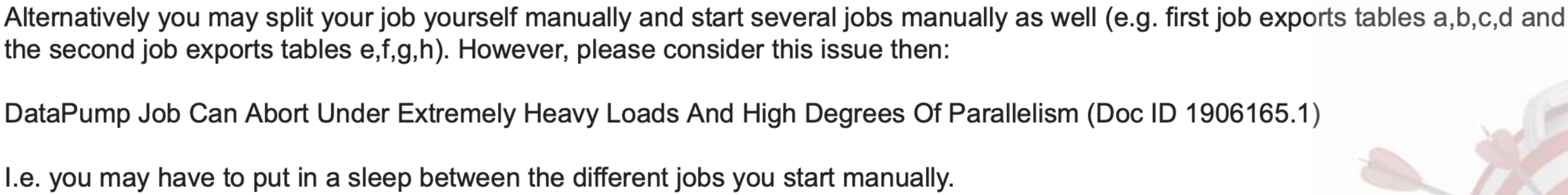
Approach
Problem resolution process

- What is the root cause?
- → Outcome of Analysis

- How can I change the approach/process/system to resolve the issue permanently?
- → Solution



Unfortunately we can't update the database and I need to continue exporting... I don't know what I'm going to do. Maybe restarting the database would do some good as it's been up since December 2023...



Alternatively you may split your job yourself manually and start several jobs manually as well (e.g. first job exports tables a,b,c,d and the second job exports tables e,f,g,h). However, please consider this issue then:

DataPump Job Can Abort Under Extremely Heavy Loads And High Degrees Of Parallelism (Doc ID 1906165.1)

I.e. you may have to put in a sleep between the different jobs you start manually.

Approach
Problem resolution process

- IMPORTANT at every step:
- CHECK THE DATA YOU HAVE: BE FOCUSED AND CHECK CAREFULLY!!!

- The 4th argument of an ORA-600 may be important
- The important information in a trace file may not be at the beginning
- Knowing the context you are in helps!
  - What version of the DB exactly?
  - What version of the client exactly?
  - Is it failing during DML or DDL?
  - What statement caused it?
  - If it’s RAC then it may have happened on the other instance!

Approach
Where to find the needed data or how to get more info?

Tools for the Analysis

- alert.log (system, db, grid infrastructure)
- trace files
- generate trace files to get more data (e.g. the errorstack)
  - set events to get an errorstack or a sql-trace
  - oradebug to e.g. get a short stack

Approach
Where to find the needed data or how to get more info?

Tools for the Analysis

- Oracle dictionary
  - dba-views
  - V$-views
- Audit information
- Autonomous Health Framework

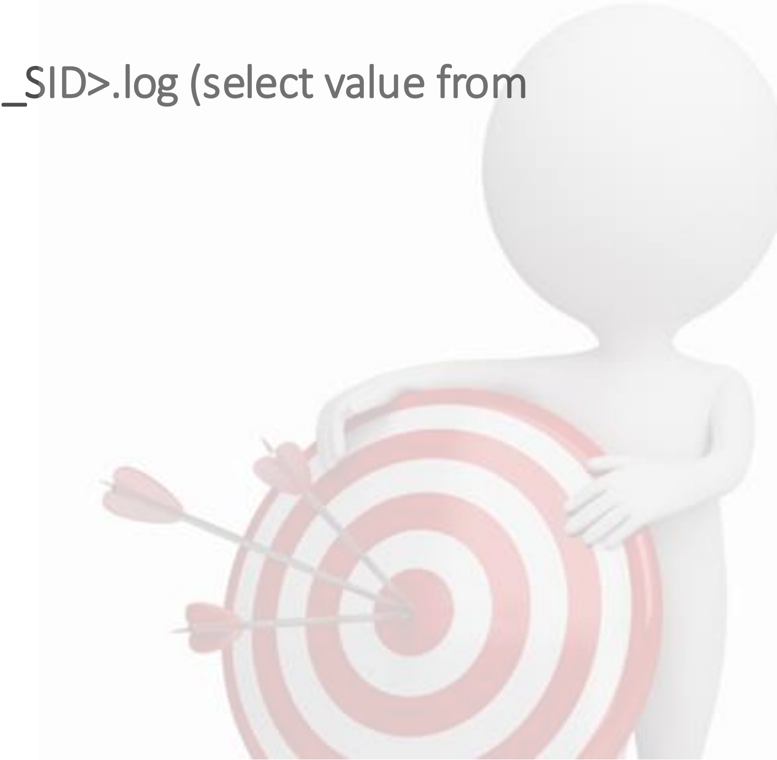
Approach
Where to find the needed data or how to get more info?

Tools for the Analysis: alert.log and trace files

- System:
  - files in/var/log, e.g. /var/log/messages
- DB: ADR = Automatic Diagnostic Repository
  - <ADR_BASE>/diag/rdbms/<DB_UNIQUE_NAME>/<ORACLE_SID>/trace/alert<ORACLE_SID>.log (select value from v$diag_info where name='Diag Trace')
- Grid Infrastructure:
  - <ADR_BASE>/diag/crs/<hostname>/crs/trace/alert.log

- Use adrci (ADR Command Interpreter) to find resources if in doubt!

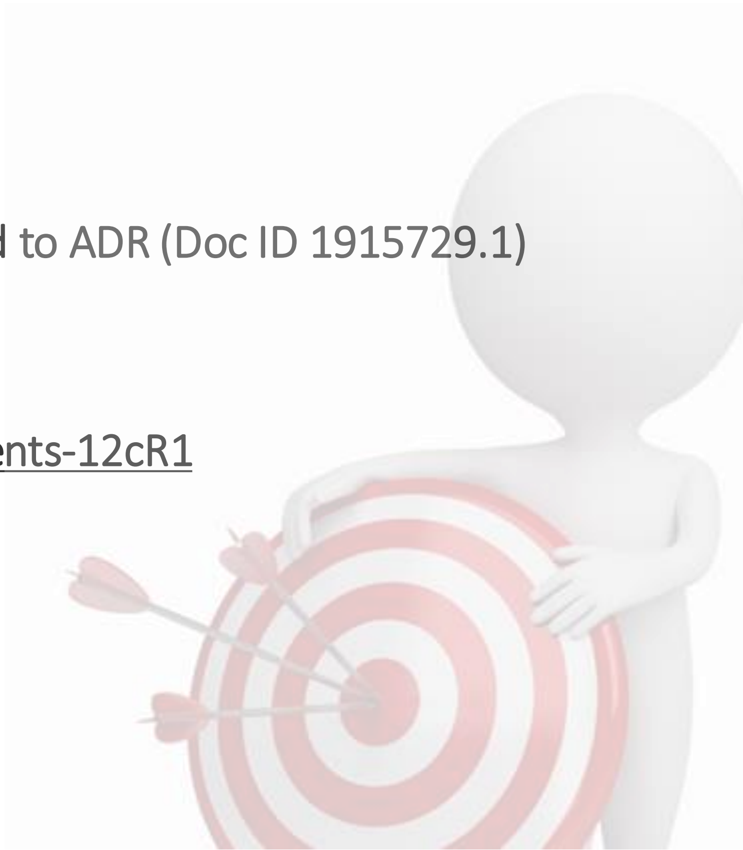
- Trace-files are also in the directory where the alert.log resides.



Approach
Where to find the needed data or how to get more info?

Tools for the Analysis: alert.log and trace files

- Resources
- MOS
  - Primary Note for Diagnosability - ADR and Packaging (Doc ID 1283137.1)
  - Database Manageability: Diagnosibility archived Webcasts (Doc ID 1268733.1)
  - 12.1.0.2 Grid Infrastructure Oracle Clusterware Diagnostic (traces) and Alert Log Moved to ADR (Doc ID 1915729.1)
- ORACLE_BASE
  - https://oracle-base.com/articles/11g/automatic-diagnostics-repository-11gr1
  - https://oracle-base.com/articles/12c/automatic-diagnostics-repository-adr-enhancements-12cR1



Approach

What about Performance issues ???

Approach
Performance issues

It’s all about TIME

- What process or sql-statement do we lose TIME on?
- Where is TIME going?
- What is the reason for losing the TIME?
- How can I change the approach/process/sql-statement to reduce TIME?

That approach can be used for ALL tuning activities (slow SQL, Locking, Mutex-Waits, IO-issue, etc.)

Approach
Performance issues

It’s all about TIME:

- What process or sql-statement do we lose TIME on? What is the problem?
- → Problem description
  - Batch takes too long
  - Order entry takes too long
  - “Overall” slowness
  - Slow DML-, DDL-, Commit-Statement

- Where is the TIME going?
- → Analysis
  - DB: AWR/Statspack-Report, Active Session History
  - Session: 10046-Trace, Active Session History
  - SQL-Statement: 10046-Trace, Active Session History, Monitoring

Approach
Performance issues

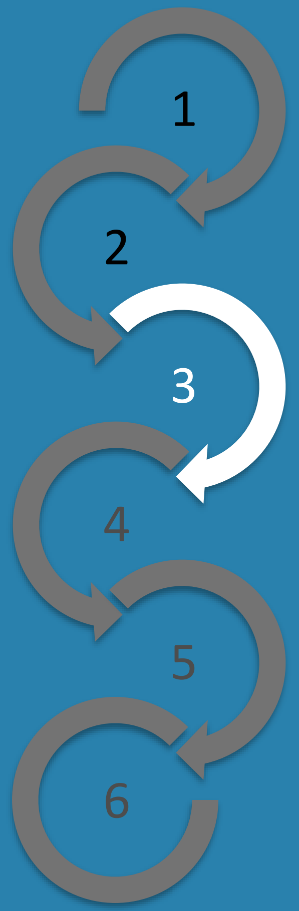
It’s all about TIME

- What is the reason for losing the TIME?
- → Outcome of Analysis
  - Bad plan (too many IOs, too much CPU-time in a heavy nested loop)
  - Slow IO-subsystem
  - Mutex-waits (no cursor sharing)
  - Slow DML due to too many triggers

- How can I change the approach/process/sql-statement to reduce TIME?
- → Solution
  - Rewrite SQL-statement
  - Use SSDs instead of HDDs
  - Use Bind variables
  - Change DML instead of using Triggers

Case 1

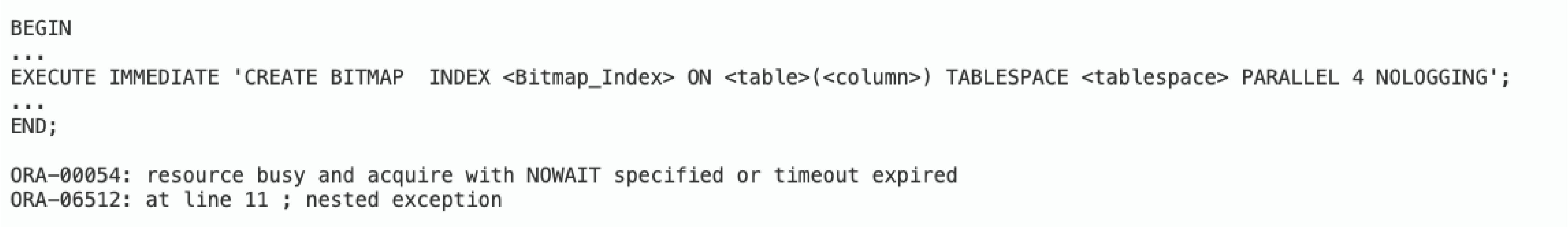
> ORA-00054: resource busy



Case 1
ORA-00054: resource busy

> Problem description:

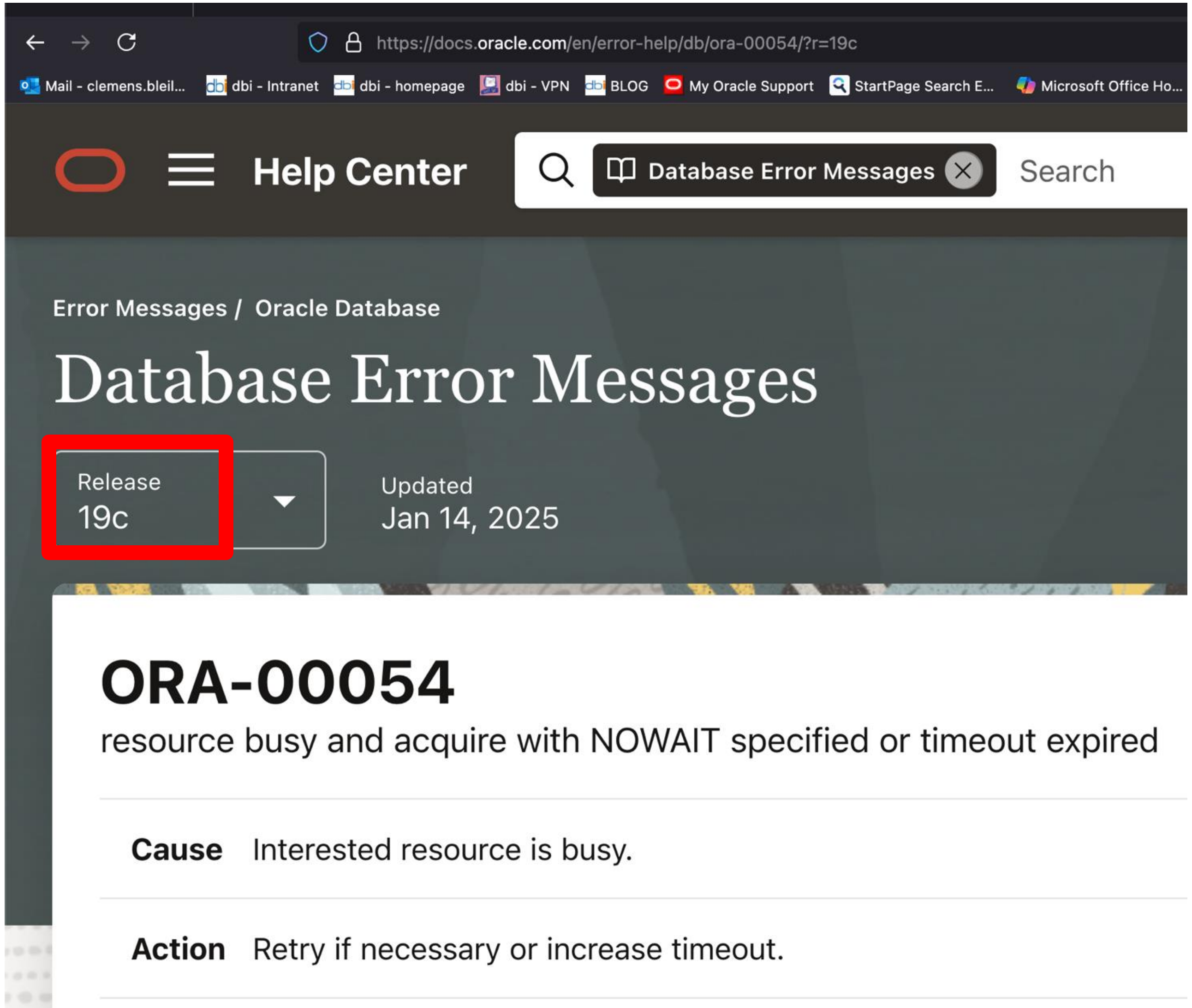
> ETL job failed on an Exadata Cloud @ Customer:



```
BEGIN
...
EXECUTE IMMEDIATE 'CREATE BITMAP  INDEX <Bitmap_Index> ON <table>(<column>) TABLESPACE <tablespace> PARALLEL 4 NOLOGGING';
...
END;

ORA-00054: resource busy and acquire with NOWAIT specified or timeout expired
ORA-06512: at line 11 ; nested exception
```

- > Disabling Parallelism worked around the issue.
- > This did not happen on the old plattform (Oracle on-prem on Linux, single instance).

Case 1
ORA-00054: resource busy



Error Messages / Oracle Database

Database Error Messages

Release 19c

Updated Jan 14, 2025

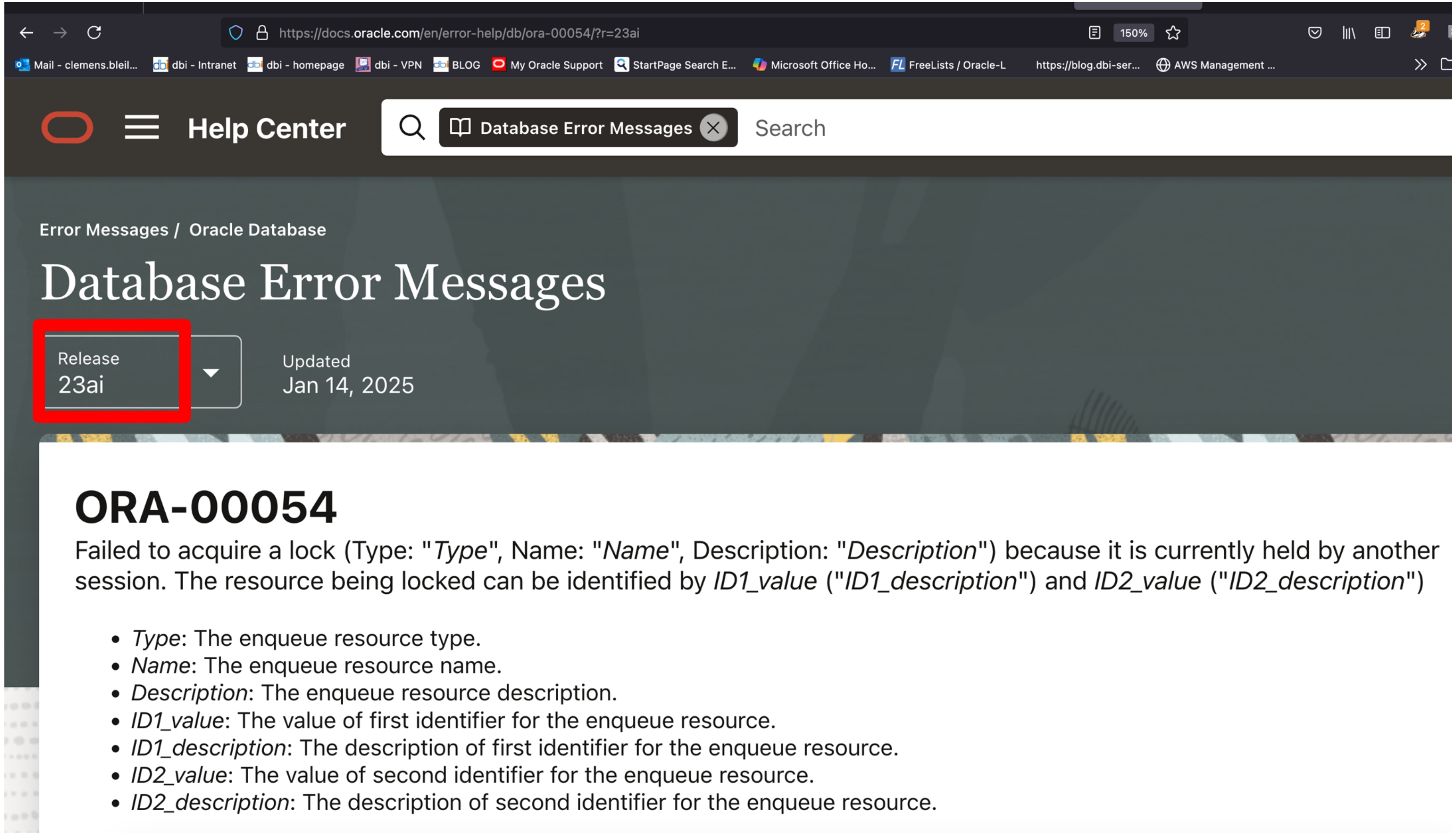
ORA-00054

resource busy and acquire with NOWAIT specified or timeout expired

Cause Interested resource is busy.

Action Retry if necessary or increase timeout.

Case 1
ORA-00054: resource busy



Error Messages / Oracle Database

Database Error Messages

Release
23ai

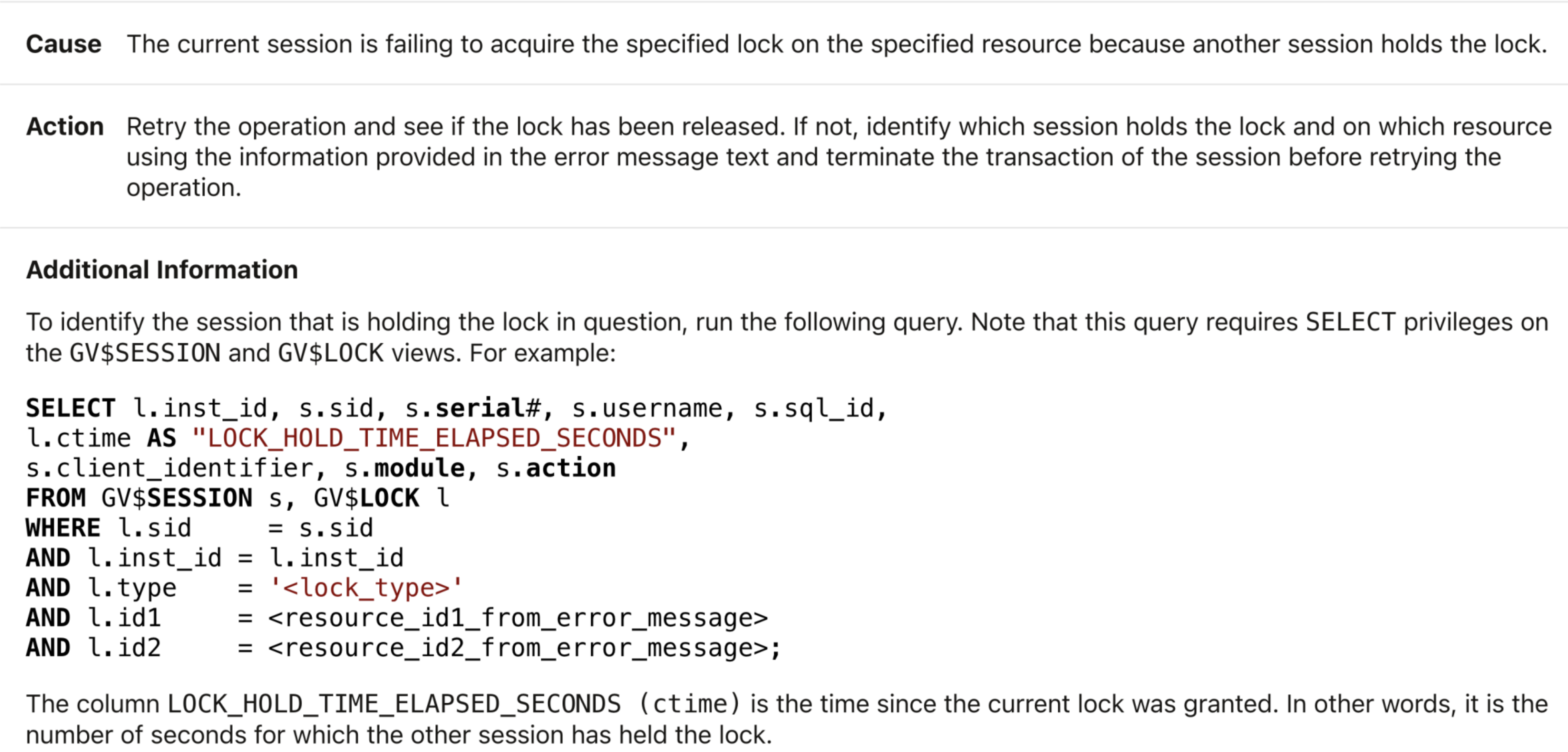
Updated
Jan 14, 2025

ORA-00054

Failed to acquire a lock (Type: "*Type*", Name: "*Name*", Description: "*Description*") because it is currently held by another session. The resource being locked can be identified by *ID1_value* ("*ID1_description*") and *ID2_value* ("*ID2_description*")

- *Type*: The enqueue resource type.
- *Name*: The enqueue resource name.
- *Description*: The enqueue resource description.
- *ID1_value*: The value of first identifier for the enqueue resource.
- *ID1_description*: The description of first identifier for the enqueue resource.
- *ID2_value*: The value of second identifier for the enqueue resource.
- *ID2_description*: The description of second identifier for the enqueue resource.

Case 1
ORA-00054: resource busy



Cause The current session is failing to acquire the specified lock on the specified resource because another session holds the lock.

Action Retry the operation and see if the lock has been released. If not, identify which session holds the lock and on which resource using the information provided in the error message text and terminate the transaction of the session before retrying the operation.

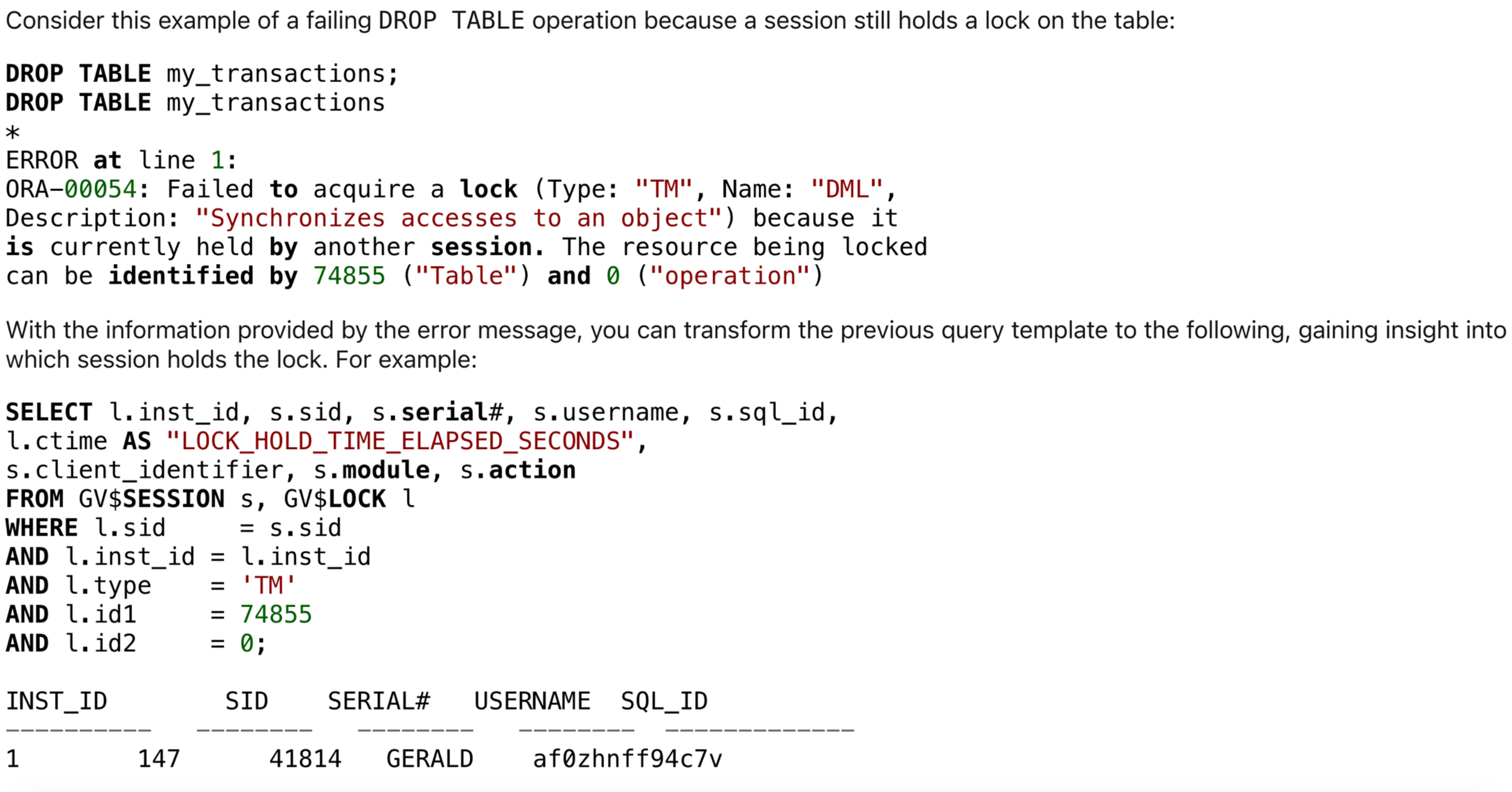
Additional Information

To identify the session that is holding the lock in question, run the following query. Note that this query requires SELECT privileges on the GV$SESSION and GV$LOCK views. For example:

```
SELECT l.inst_id, s.sid, s.serial#, s.username, s.sql_id,
l.ctime AS "LOCK_HOLD_TIME_ELAPSED_SECONDS",
s.client_identifier, s.module, s.action
FROM GV$SESSION s, GV$LOCK l
WHERE l.sid     = s.sid
AND l.inst_id = l.inst_id
AND l.type    = '<lock_type>'
AND l.id1     = <resource_id1_from_error_message>
AND l.id2     = <resource_id2_from_error_message>;
```

The column LOCK_HOLD_TIME_ELAPSED_SECONDS (ctime) is the time since the current lock was granted. In other words, it is the number of seconds for which the other session has held the lock.

Case 1
ORA-00054: resource busy



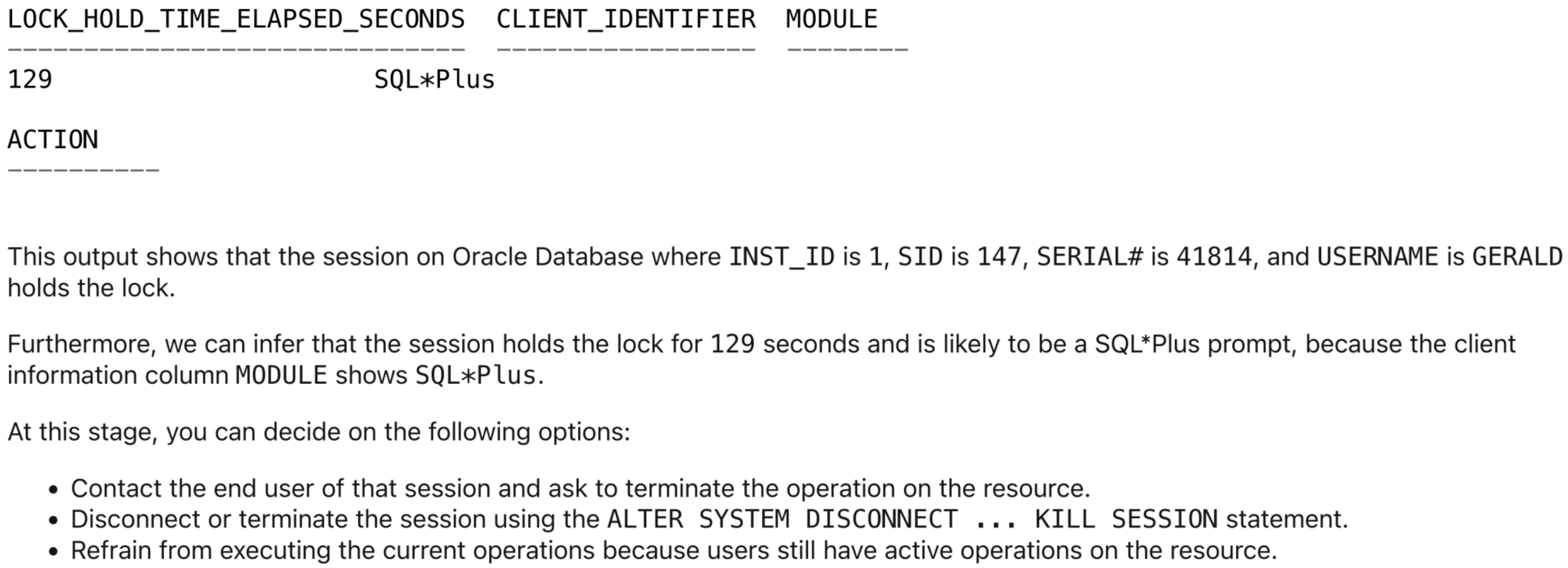
Consider this example of a failing DROP TABLE operation because a session still holds a lock on the table:

```
DROP TABLE my_transactions;
DROP TABLE my_transactions
*
ERROR at line 1:
ORA-00054: Failed to acquire a lock (Type: "TM", Name: "DML",
Description: "Synchronizes accesses to an object") because it
is currently held by another session. The resource being locked
can be identified by 74855 ("Table") and 0 ("operation")
```

With the information provided by the error message, you can transform the previous query template to the following, gaining insight into which session holds the lock. For example:

```
SELECT l.inst_id, s.sid, s.serial#, s.username, s.sql_id,
l.ctime AS "LOCK_HOLD_TIME_ELAPSED_SECONDS",
s.client_identifier, s.module, s.action
FROM GV$SESSION s, GV$LOCK l
WHERE l.sid     = s.sid
AND l.inst_id = l.inst_id
AND l.type    = 'TM'
AND l.id1     = 74855
AND l.id2     = 0;

INST_ID        SID    SERIAL#   USERNAME  SQL_ID
----------   --------   --------   --------  -------------
1        147       41814    GERALD     af0zhnff94c7v
```

Case 1
ORA-00054: resource busy



```
LOCK_HOLD_TIME_ELAPSED_SECONDS  CLIENT_IDENTIFIER  MODULE
------------------------------  -----------------  --------
129                     SQL*Plus

ACTION
----------
```

This output shows that the session on Oracle Database where INST_ID is 1, SID is 147, SERIAL# is 41814, and USERNAME is GERALD holds the lock.

Furthermore, we can infer that the session holds the lock for 129 seconds and is likely to be a SQL*Plus prompt, because the client information column MODULE shows SQL*Plus.

At this stage, you can decide on the following options:

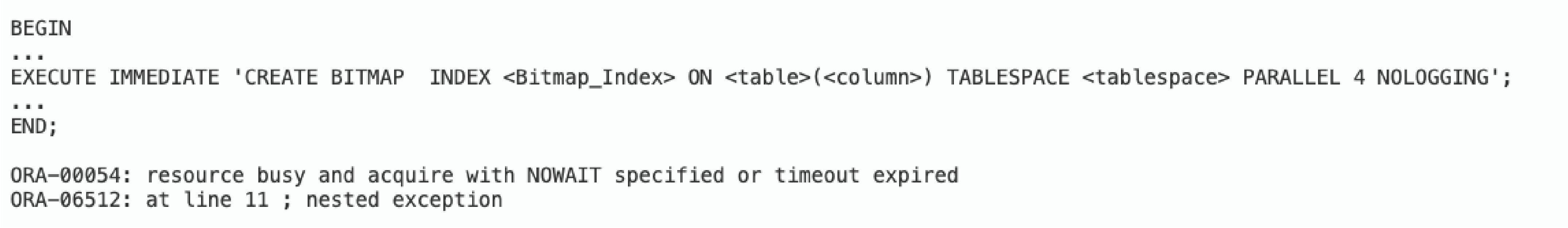
- Contact the end user of that session and ask to terminate the operation on the resource.
- Disconnect or terminate the session using the ALTER SYSTEM DISCONNECT ... KILL SESSION statement.
- Refrain from executing the current operations because users still have active operations on the resource.

Case 1
ORA-00054: resource busy

- ORA-00054 example

Case 1
ORA-00054: resource busy

> From the problem description:



```
BEGIN
...
EXECUTE IMMEDIATE 'CREATE BITMAP  INDEX <Bitmap_Index> ON <table>(<column>) TABLESPACE <tablespace> PARALLEL 4 NOLOGGING';
...
END;

ORA-00054: resource busy and acquire with NOWAIT specified or timeout expired
ORA-06512: at line 11 ; nested exception
```

Case 1
ORA-00054: resource busy

> Analysis

> No trace file generated with an ORA-00054

> To get more information I enabled to produce an errorstack when the issue happens again:

```
SQL> alter system set events '54 trace name errorstack forever, level 1’;

Check that the event has been set:
SQL> oradebug setmypid
Statement processed.
SQL> oradebug eventdump session
54 trace name errorstack forever, level 1
SQL>
```

> REMARK: Enable this on both instances as there is no «sid=‘*’» syntax.

Case 1
ORA-00054: resource busy

> Analysis

> After reproducing the issue a trace file has been created:

```
dbkedDefDump(): Starting a non-incident diagnostic dump (flags=0x0, level=1, mask=0x0)
----- Error Stack Dump -----
<error barrier> at 0x7ffe8ccd6720 placed dbkda.c@298
ORA-00054: resource busy and acquire with NOWAIT specified or timeout expired
----- Current SQL Statement for this session (sql_id=gs325wgqdv7zm) -----
LOCK TABLE ”<table-name>" IN SHARE MODE  NOWAIT
….
dbkedDefDump(): Starting a non-incident diagnostic dump (flags=0x0, level=1, mask=0x0)
----- Error Stack Dump -----
<error barrier> at 0x7ffe8ccd9c80 placed dbkda.c@298
ORA-00054: resource busy and acquire with NOWAIT specified or timeout expired
----- Current SQL Statement for this session (sql_id=7nsznp02kwwg7) -----
CREATE BITMAP
INDEX <bitmap-index> ON <table-name>(<column-name>)
 TABLESPACE <ts-name> PARALLEL 4 NOLOGGING
….
```

Case 1
ORA-00054: resource busy

> Analysis
> Checked ASH to see a blocking session:

```
SQL> select count(*) from gv$active_session_history where sql_id='7nsznp02kwwg7';

  COUNT(*)
----------
        57

SQL>  select distinct blocking_session from gv$active session_history where sql_id='7nsznp02kwwg7';

BLOCKING_SESSION
----------------

                                            NO

SQL> select inst_id, session_state, event              om gv$active_session_history where
sql_id='7nsznp02kwwg7' group by inst_id, session_state, event order by 4;

   INST_ID SESSION EVENT                                                              COUNT(*)
---------- ------- ---------------------------------------------------------------- ----------
…
         2 WAITING cell single block physical read: flash cache                             11
         2 ON CPU                                                                            41
```

Case 1
ORA-00054: resource busy

- Analysis
- Asked the customer to re-run the job after increasing DDL_LOCK_TIMEOUT to 5 mins:

```
SQL> show spparameter ddl_lock_timeout

SID      NAME                          TYPE        VALUE
-------- ----------------------------- ----------- ----------------------------
*        ddl_lock_timeout              integer
SQL> alter system set ddl_lock_timeout=300 sid='*';

Syste

SQL>

SID
-----
*        ddl_lock_timeout              integer     300
SQL> show parameter ddl_lock_timeout

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
ddl_lock_timeout                     integer     300
```

Still no data in ASH

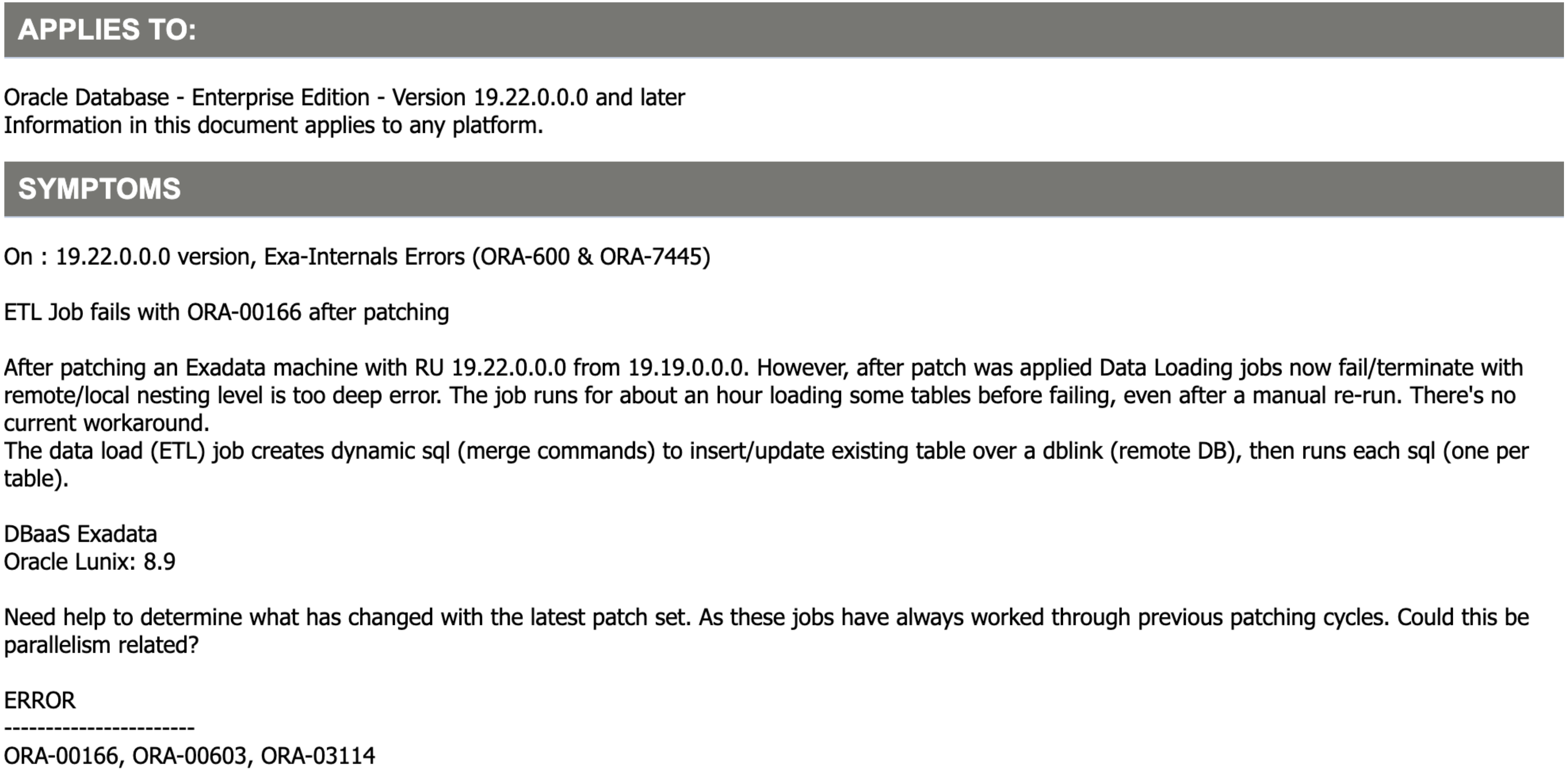
Case 1
ORA-00054: resource busy

- Analysis
- A trace file was produced a while before the ORA-00054 happended:

```
ORA-00603: ORACLE server session terminated by fatal error
ORA-00166: remote/local nesting level is too deep
ORA-00166: remote/local nesting level is too deep

…
----- Current SQL Statement for this session (sql_id=7cq4f99zx3gh4) -----

INSERT /*+ ENABLE_PARALLEL_DML PARALLEL(4) */ INTO <table-name>(
…
```

Case 1
ORA-00054: resource busy

> MOS Note: ETL Job Fails With ORA-00166 After Patching (Doc ID 3025230.1)



APPLIES TO:

Oracle Database - Enterprise Edition - Version 19.22.0.0.0 and later
Information in this document applies to any platform.

SYMPTOMS

On : 19.22.0.0.0 version, Exa-Internals Errors (ORA-600 & ORA-7445)

ETL Job fails with ORA-00166 after patching

After patching an Exadata machine with RU 19.22.0.0.0 from 19.19.0.0.0. However, after patch was applied Data Loading jobs now fail/terminate with remote/local nesting level is too deep error. The job runs for about an hour loading some tables before failing, even after a manual re-run. There's no current workaround.
The data load (ETL) job creates dynamic sql (merge commands) to insert/update existing table over a dblink (remote DB), then runs each sql (one per table).

DBaaS Exadata
Oracle Lunix: 8.9

Need help to determine what has changed with the latest patch set. As these jobs have always worked through previous patching cycles. Could this be parallelism related?

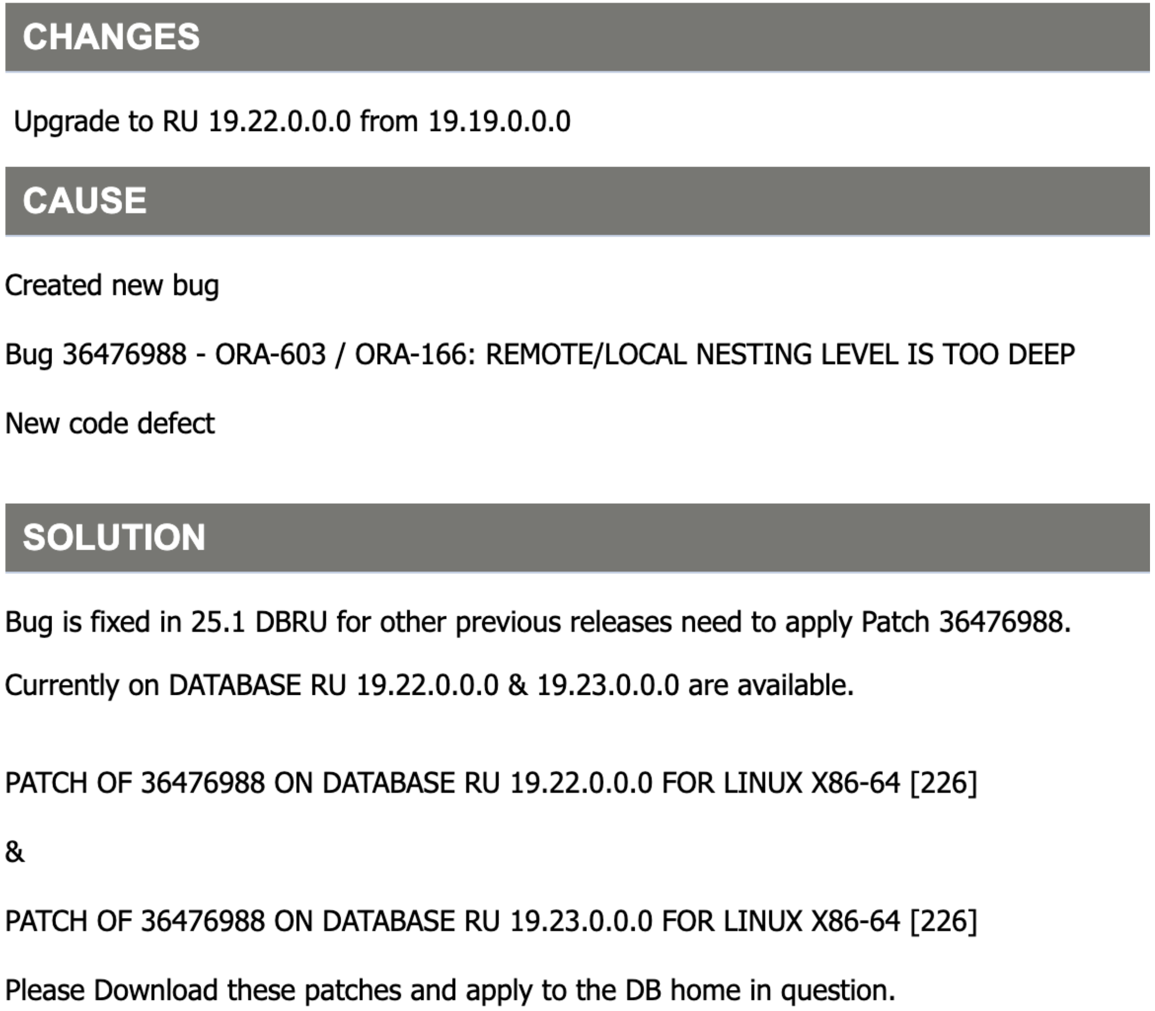
ERROR

ORA-00166, ORA-00603, ORA-03114

Case 1

ORA-00054: resource busy

> MOS Note: ETL Job Fails With ORA-00166 After Patching (Doc ID 3025230.1)



CHANGES

Upgrade to RU 19.22.0.0.0 from 19.19.0.0.0

CAUSE

Created new bug

Bug 36476988 - ORA-603 / ORA-166: REMOTE/LOCAL NESTING LEVEL IS TOO DEEP

New code defect

SOLUTION

Bug is fixed in 25.1 DBRU for other previous releases need to apply Patch 36476988.

Currently on DATABASE RU 19.22.0.0.0 & 19.23.0.0.0 are available.

PATCH OF 36476988 ON DATABASE RU 19.22.0.0.0 FOR LINUX X86-64 [226]

&

PATCH OF 36476988 ON DATABASE RU 19.23.0.0.0 FOR LINUX X86-64 [226]

Please Download these patches and apply to the DB home in question.

Case 1
ORA-00054: resource busy

> Result of the Analysis

> A session made a DML on a table and failed after some time with error

```
ORA-00603: ORACLE server session terminated by fatal error
ORA-00166: remote/local nesting level is too deep
ORA-00166: remote/local nesting level is too deep

…
----- Current SQL Statement for this session (sql_id=7cq4f99zx3gh4) -----

INSERT /*+ ENABLE_PARALLEL_DML PARALLEL(4) */ INTO <table-name>(
…
```

Case 1

ORA-00054: resource busy

> Solution

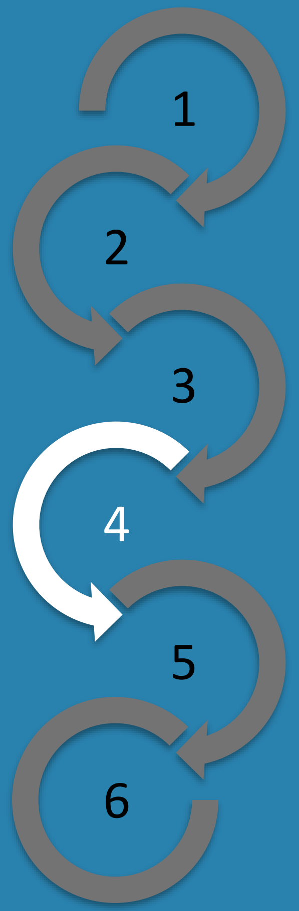
> Apply patch for bug 36476988 on our 19.25.

> The bug is fixed in 19.26.

> As this error only happened in one PDB only, I created a new 19.25.-ORACLE_HOME with the patch applied, created a CDB there and relocated the PDB with the issue to that CDB

Case 2

- > The issue
- > Analysis
- > Output of analysis
- > Solution

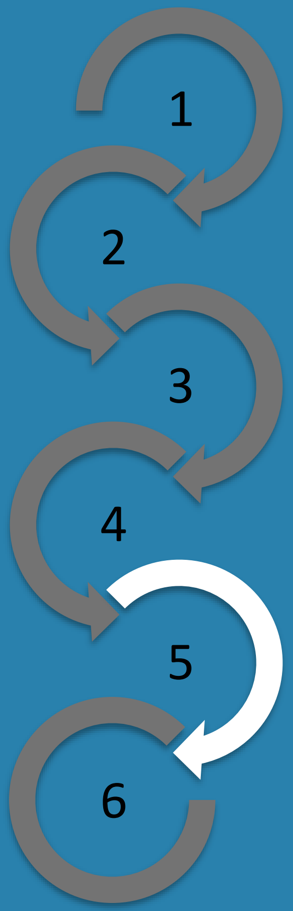


Case 2
The issue

> See separate file due to customer specific data

Tools available

- oradebug
- events
- orafun.info
- performance tools



Tools available
oradebug

Use for very specific cases only (ask Oracle Support if in doubt)

- oradebug help
- oradebug setospid <ospid>
- oradebug setorapid <orapid>
  - <orapid> is the pid from v$process
- oradebug unlimit
- oradebug settracefileid clemens
- oradebug EVENT 10046 TRACE NAME CONTEXT FOREVER, LEVEL 12
- oradebug EVENT 10046 TRACE NAME CONTEXT OFF
- oradebug short_stack
- oradebug suspend
- oradebug wakeup <orapid>

Tools available
events

Set an event for a future occurrence of it without setting it persistently

- alter system set events ’<ORA-Error-Number> trace name errorstack forever, level 1’;

- alter system set events ’942 trace name errorstack forever, level 1’;
- alter system set events ’942 trace name errorstack off’;

- alter system set events 'sql_trace[sql: cjrha4bzuupzf] level=12’;
- https://www.dbi-services.com/blog/enable-10046-tracing-for-a-specific-sql/

- How to see what events have been set?
- oradebug setmypid
- oradebug eventdump session

Tools available
orafun.info

http://orafun.info/ created by Frits Hoogland with a little help from Kawil Stawiarski

→ Get more info about function names in the stack

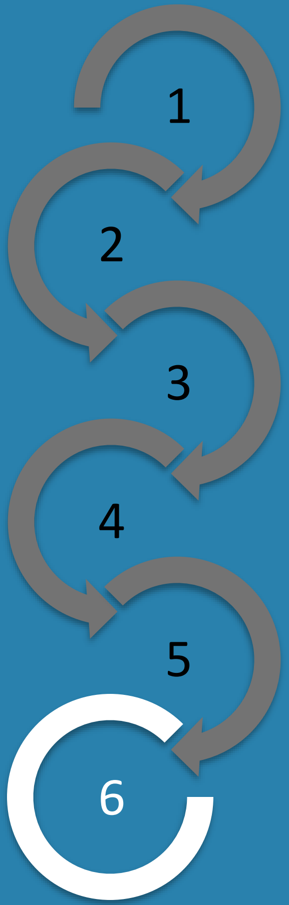
Tools available
Performance tools

- ASH
- AWR/Statspack
- SQL_TRACE
- Tanel Poder’s Snapper
- Third party tools like d.side for Standard Edition 2 DBs

Recommendations and Summary

> Recommendations

> Summary



Recommendations

- Do a proper **analysis yourself** before opening a SR
- Become familiar with **tools**, so that you are confident when using them: “**Be prepared**”
- Read the data you get **carefully**
- **Talk to colleagues and on conferences** about issues you faced
- When searching in MOS: **Use modules in the upper part of the stack** (the ones that appear below the error handling routines: e.g. below kge..)
- Have a **test DB available on your Laptop or on your test server** to reproduce issues

Summary

Summary

- You can resolve a lot of issues **yourself**

- Most of the time **much faster** than Oracle Support

- This saves your organization a lot of time and money

- If you are dealing with a sev1 issue or if you are stuck, then open a SR of course

- Try to use a scientific approach and verify your findings on a test system

- Do it like
- the Die Autodoktoren

QUESTIONS?

Please do ask

GETTING
GREAT
PEOPLE
TOGETHER