HERZLICH WILLKOMMEN

Making of: Der Preis den wir zahlen – Performance und HA mit Oracle RAC

Axel vom Stein

mit Martin Klier  **Performing Databases GmbH**

- Technischer Projektleiter, Leitung Datentechnik (> 20 Jahre BSS Bohnenberg GmbH)

- Oracle:
  - seit 2001
  - zunächst als Entwickler (PL/SQL, C, C#)
  - dann mehr und mehr Richtung DBA

- Schwerpunkte/Faible:
  - Ausbildung
  - Sicherheits- und HA - Konzepte
  - Migrationsprojekte
  - Standardisierung
  - alles was sich automatisieren lässt

Unternehmen – Fakten …

Hauptsitz: Solingen (NRW)

Standorte: Kattowitz (Polen), Wels (Österreich), Ilmenau

Gegründet: 1991

Rechtsform: GmbH

seit 2006: Unternehmen der ROFA Industrial Automation Group

Mitarbeiter: ca. 170

- Entwurf und Feinplanung von intralogistischen Systemen und Anlagen
- schlüsselfertiger Abwicklung als GU
- Flowpicker (Hochleistungskommissionierung)
- Hochregal- und Kleinteilelagern
- fahrerlosen Transportsystemen
- Lagerverwaltungssystemen
- Materialflussrechnern
- Pick-by-Light / Pick-by-Voice
- Staplerleitsystemen
- Visualisierungen
- SPS – Technik
- etc.

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

SYMPOSIUM 42
Proud Member of symposium42

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

Agenda …

1.) Motivation

2.) Herausforderungen

3.) Tools

4.) Zusammenfassung

- mehrjährige Zusammenarbeit in Projekten des Logistikumfeldes
- Auf der DB-Konferenz: Perfomance und HA mit Data Guard
- CfP für K&A endet kurz vor der DB-Konferenz
- Vortrag für DB-Konferenz hat Spaß gemacht
- wir haben ja schon Material auf dem wir Aufsetzen können …

- Neben der technischer Neugierde aber auch eine Antwort auf:

Ist RAC sinnvoll für die jetzige BSS-Applikation?

- ein wenig mehr „on site“ als beim letzten Mal wäre sinnvoll
- offen sein für Abweichungen vom Plan
- jederzeit mit Überraschungen rechnen
- manchmal ist weniger wirklich mehr
- mehr Zeit für Messreihen vorsehen
- keine Virtualisierung, sondern Hardware

1.) Motivation

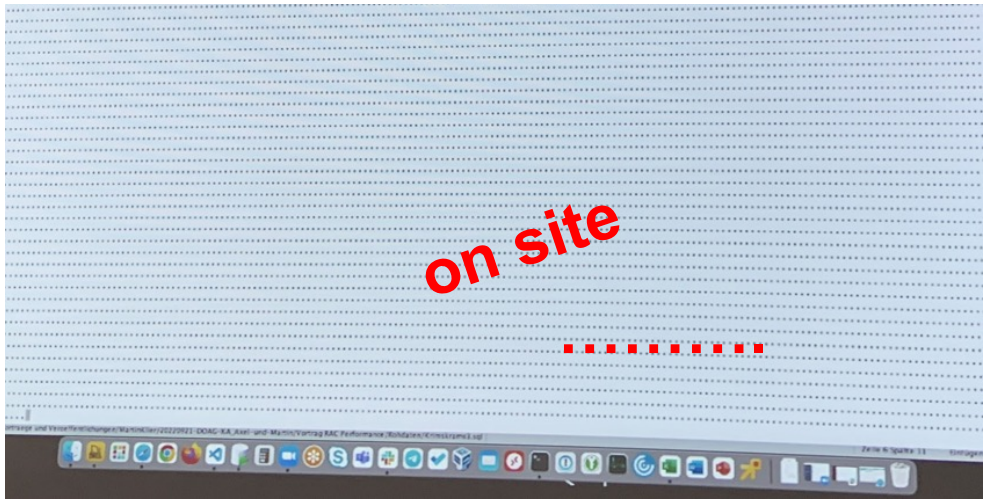
2.) Herausforderungen

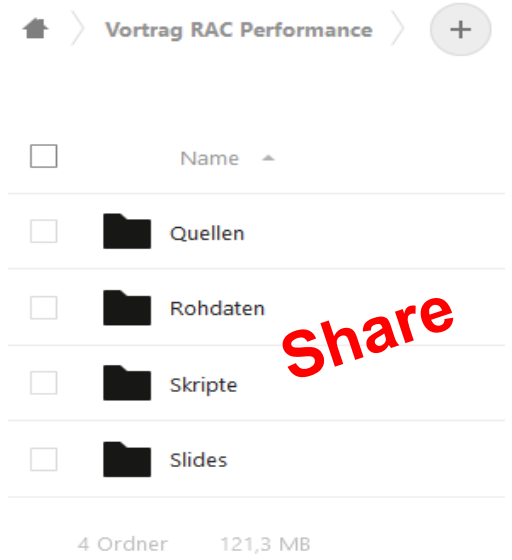
3.) Tools

4.) Zusammenfassung

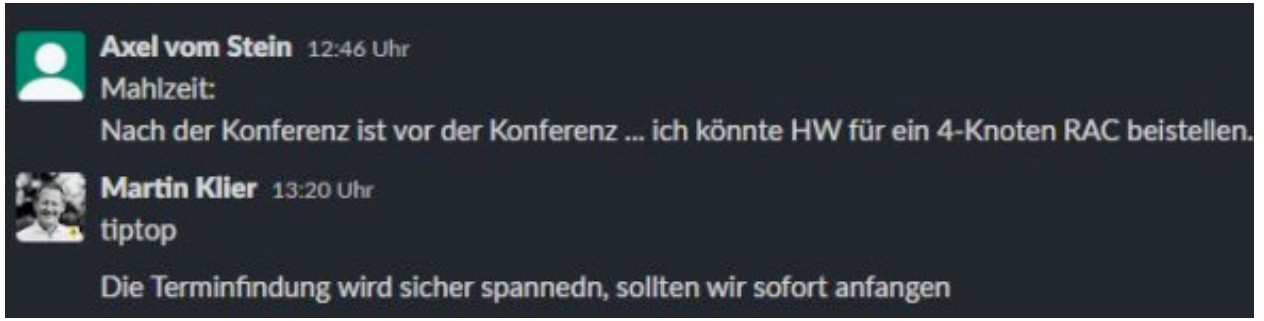

Solingen (DE-NW)
555 km
Mitterteich (DE-BY)
Düsseldorf
Leverkusen
Köln
Bonn
Siegen
Gummersbach
Marburg
Alsfeld
Gießen
Wetzlar
Limburg an der Lahn
Koblenz
Rhein
Frankfurt am Main
Offenbach am Main
Hanau
Wiesbaden
Mainz
Darmstadt
Aschaffenburg
Bad Kreuznach
Würzburg
Fulda
Bad Hersfeld
Bebra
Eschwege
Bad Wildungen
Baunatal
Mühlhausen/ Thüringen
Bad Langensalza
Eisenach
Gotha
Erfurt
Weimar
Jena
Naumburg (Saale)
Gera
Altenburg
Zwickau
Greiz
Plauen
Ilmenau
Suhl
Meiningen
Bad Neustadt an der Saale
Bad Kissingen
Schweinfurt
Bamberg
Coburg
Bad Staffelstein
Kulmbach
Hof
Bayreuth
Marktredwitz
Eger

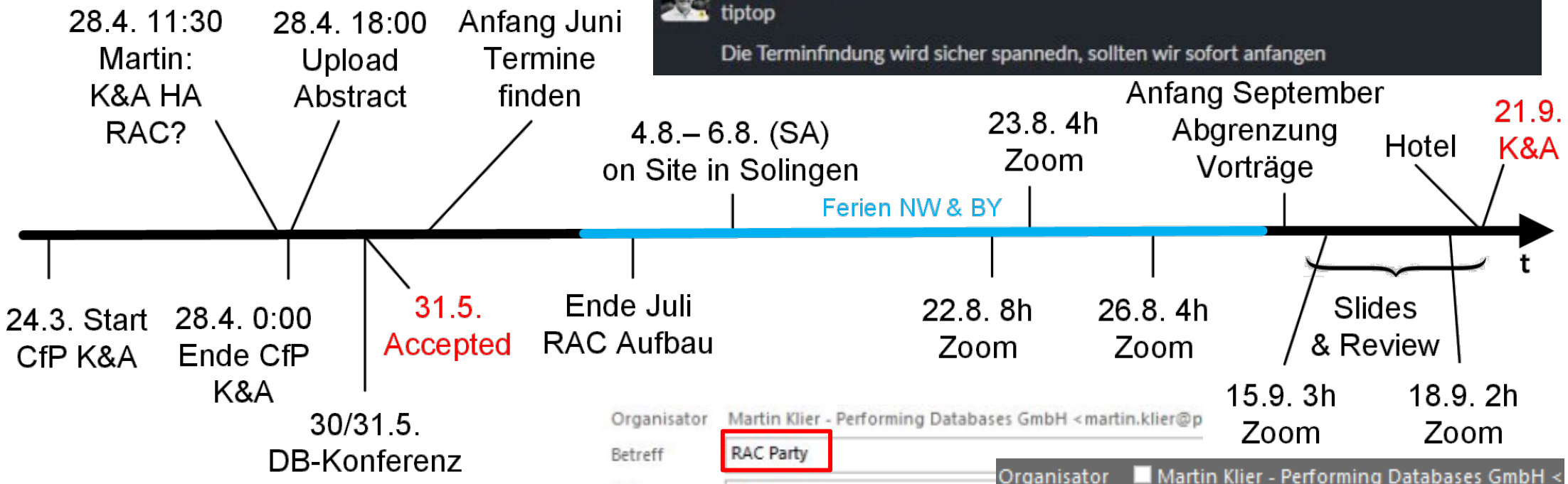
Herausforderungen: Die Distanz

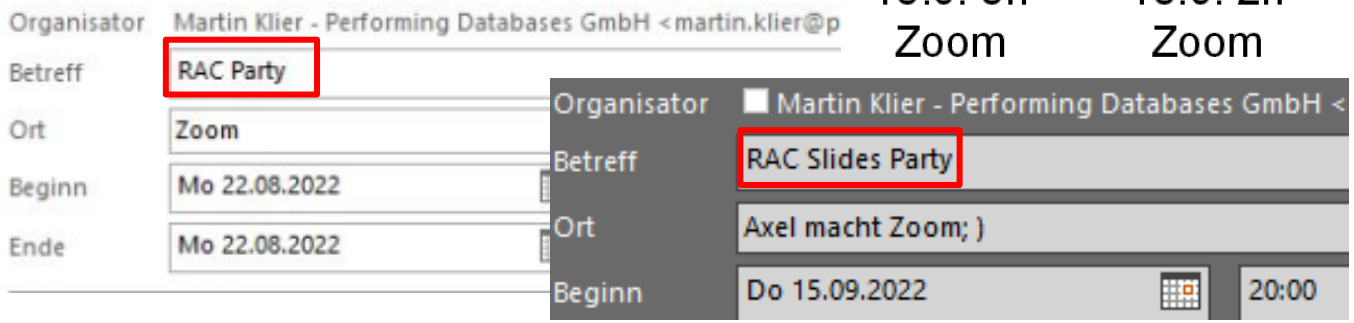


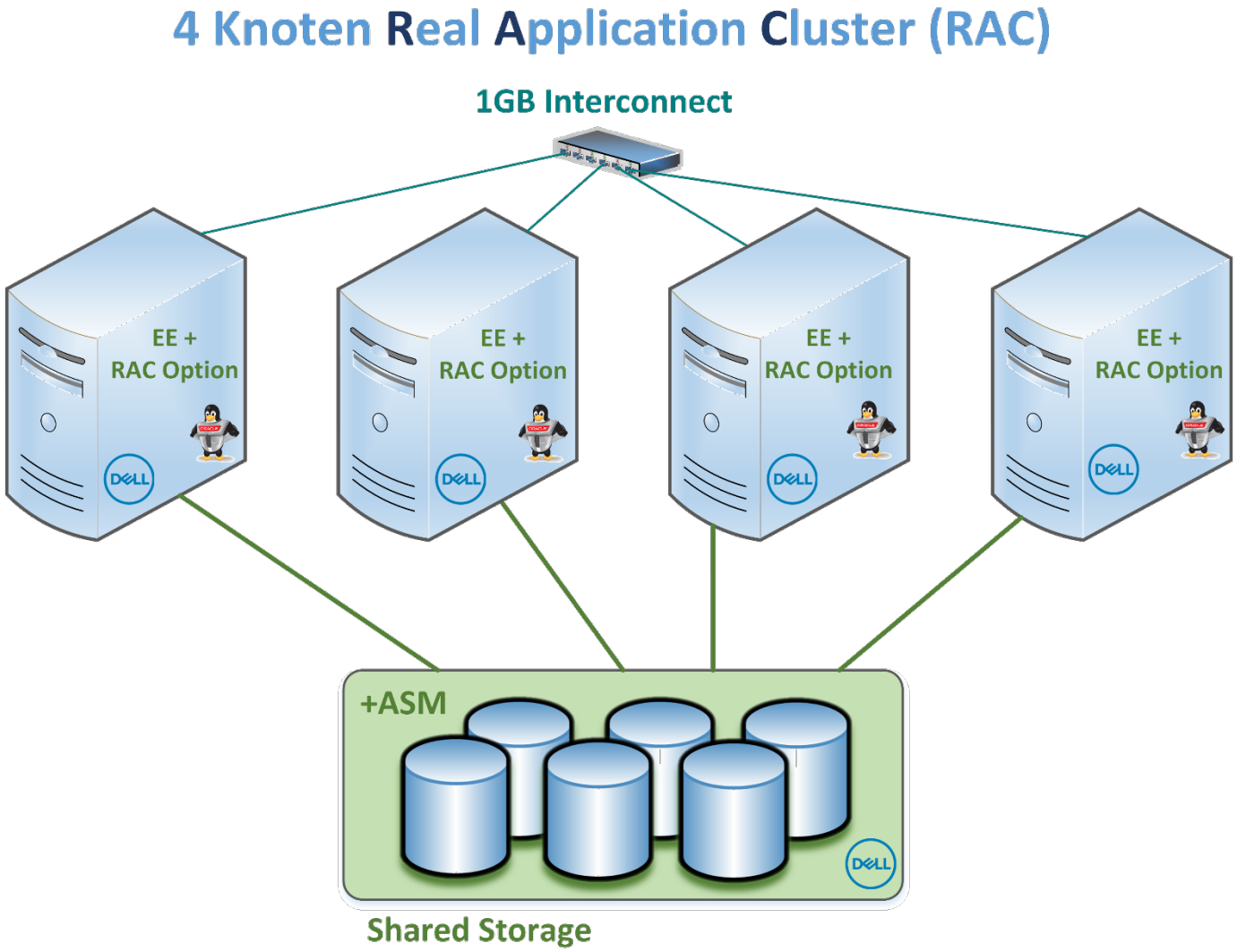


- **Zusammenarbeit bedeutet auch Planung:**

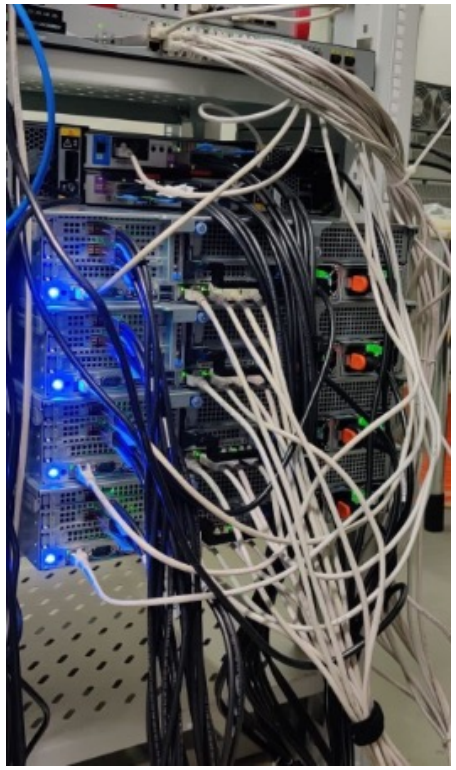







4 Knoten Real Application Cluster (RAC)
1GB Interconnect
EE +
RAC Option
EE +
RAC Option
EE +
RAC Option
EE +
RAC Option
+ASM
Shared Storage

- **HW:**
  - **4 x DELL PowerEdge R740**
  - **CPU: Intel Xeon Silver 4114 2.20GHz**
  - **RAM: 2 Rechner mit 64 GB, 2 Rechner mit 128 GB**
  - **Shared Storage PowerVault ME4024 (DAS, kein FC)**
- **OS:**
  - **Oracle Linux 7.9**
- **GRID/DB:**
  - **Grid-Version: 19.16.0.0.0 (34130714 – 220719)**
  - **DB-Version: 19.16.0.0.0 (34130714 – 220719)**
  - **Diskgroups (DATA – 1 TB, FRA – 500 GB, REDO1 – 100GB , REDO2 – 100GB)**
  - **Multitenant – Architektur mit einer PDB**

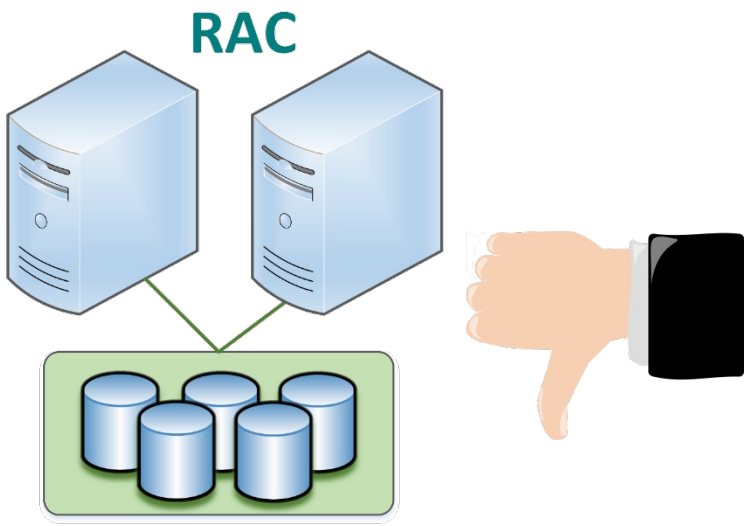


- **Auszug aus:** `crsctl stat res -t`

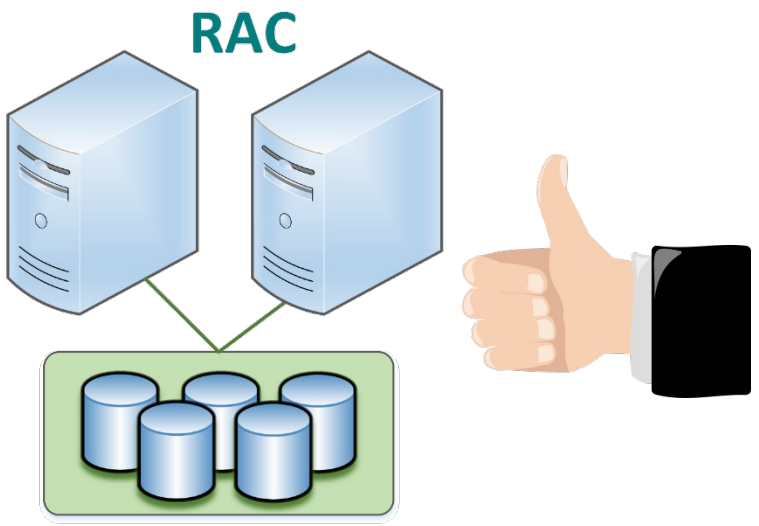
```
Instanzen:
ora.bssgta.db
                      1        ONLINE  ONLINE       bss-inhouse-ora1         Open,HOME=/oracle/19dbBSSGT      STABLE
                      2        ONLINE  ONLINE       bss-inhouse-ora2         Open,HOME=/oracle/19dbBSSGT      STABLE
                      3        ONLINE  ONLINE       bss-inhouse-ora3         Open,HOME=/oracle/19dbBSSGT      STABLE
                      4        ONLINE  ONLINE       bss-inhouse-ora4         Open,HOME=/oracle/19dbBSSGT      STABLE

Scan-Listener:
ora.GRID_SCAN4.lsnr   1        ONLINE  ONLINE       bss-inhouse-ora1                                          STABLE
ora.GRID_SCAN1.lsnr   1        ONLINE  ONLINE       bss-inhouse-ora2                                          STABLE
ora.GRID_SCAN2.lsnr   1        ONLINE  ONLINE       bss-inhouse-ora3                                          STABLE
ora.GRID_SCAN3.lsnr   1        ONLINE  ONLINE       bss-inhouse-ora4                                          STABLE

Knoten-Listener:
ora.LISTENER.lsnr              ONLINE  ONLINE       bss-inhouse-ora1                                          STABLE
                               ONLINE  ONLINE       bss-inhouse-ora2                                          STABLE
                               ONLINE  ONLINE       bss-inhouse-ora3                                          STABLE
                               ONLINE  ONLINE       bss-inhouse-ora4                                          STABLE

Services:
ora.bssgta.bssgt1p.client.svc
                      1        ONLINE  ONLINE       bss-inhouse-ora1                                          STABLE
                      2        ONLINE  ONLINE       bss-inhouse-ora2                                          STABLE
                      3        ONLINE  ONLINE       bss-inhouse-ora3                                          STABLE
                      4        ONLINE  ONLINE       bss-inhouse-ora4                                          STABLE
```

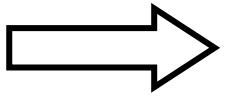
- weitere vorbereitende Dinge:
  - Nmon
  - NmonChart
  - weitere LUN `/dev/BENCHMARK1` Größe 1TB für ORION und dd

- Zudem eine VM:
  - Oracle Linux 8.3 mit GUI
  - Oracle Instant-Client / SQLPLUS
  - Swingbench
  - SQL-Developer



ODER





Natürlich Beides!

Agenda …

1.) Motivation

2.) Herausforderungen

3.) Tools

4.) Zusammenfassung

Ursprünglich glaubten wir (analog zum DG – Vortrag):

- dd
- ORION
- Excel
- Swingbench
- AWR snapshots/reports
- nmon
- nmonchart

ABER: Glauben heißt nicht …

Wir nutzten überwiegend:

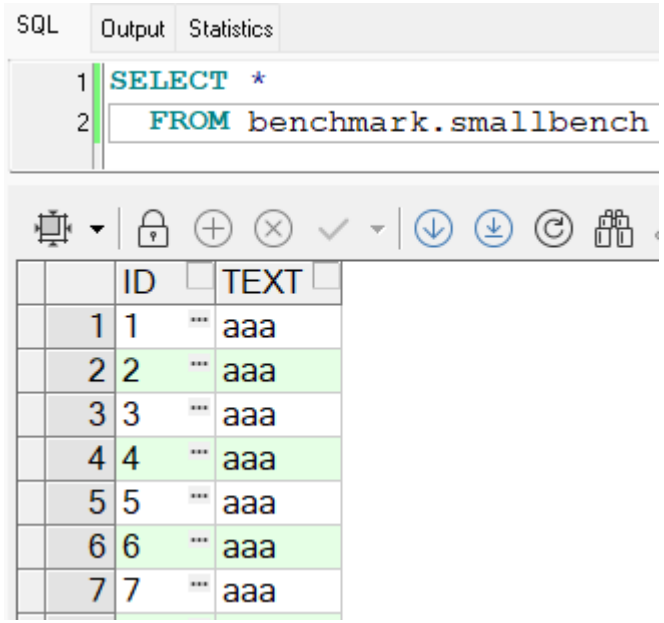
- **Bash**
- **SQL**
- **PL/SQL**
- **AWR snapshots/reports**
- **Excel**

Nur der Vollständigkeit halber:

- ~~**dd, ORION**~~
- ~~**nmonchart**~~
- **nmon**
- **Swingbench**

- Nur Swingbench schien uns ungeeignet, da nicht wirklich steuerbar → Auswirkung auf Tools

- Deshalb folgende Idee:
  - einfache Tabelle mit 2 Spalten
  - Spalte ID indiziert und Spalte Text fixer Wert
  - 1 Millionen Datensätze
  - Update / Select – Statements auf die Tabelle
  - diverse Fälle überlegen
  - pro Fall Baseline (Single Instance, bzw. 1 Knoten)
  - pro Fall Benchmark je nach Fall mit 2 bis 4 Knoten
  - AWR – Snapshots / Reports



- **Testfälle:**

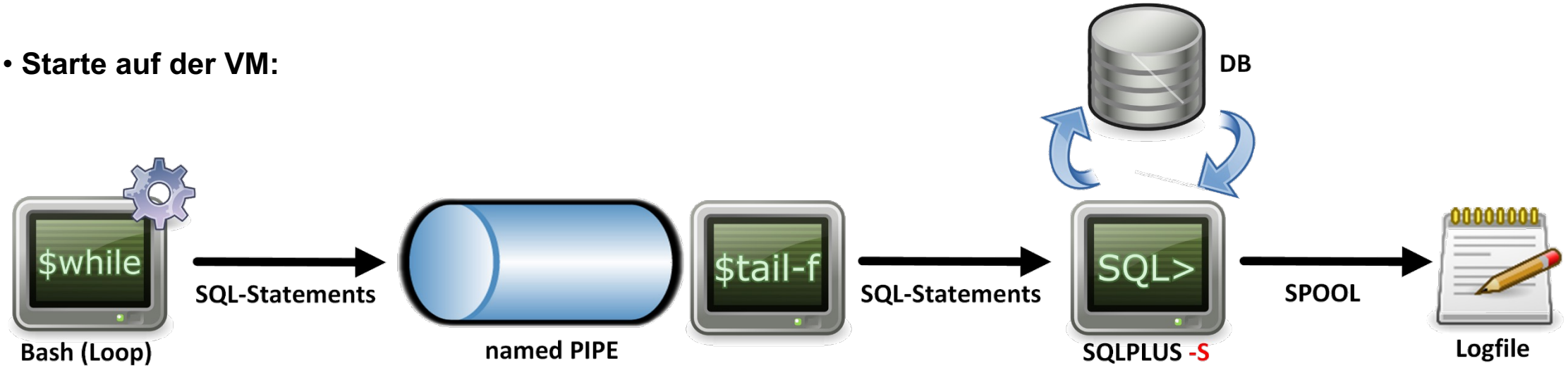
Fall	Aktion	#Knoten	#Sessions pro Knoten	ID‘s / Test
1a	SELECT	1	4	full table scan
1b	SELECT	4	1	full table scan
2a	UPDATE	1	4	erste 10.000 Datensätze bzw. 18 Blöcke
2b	UPDATE	4	1	erste 10.000 Datensätze bzw. 18 Blöcke
3a	UPDATE	1	4	4 Datensätze, Session = ID des Datensatz, d.h. alles gleicher Block
3b	UPDATE	2	2	4 Datensätze, Session = ID des Datensatz, d.h. alles gleicher Block
3c	UPDATE	4	1	4 Datensätze, Session = ID des Datensatz, d.h. alles gleicher Block
4a	UPDATE	1	4	4 Datensätze, Session = ID*10.000 des Datensatz, d.h. alles verschiedene Blöcke
4b	UPDATE	4	1	4 Datensätze, Session = ID*10.000 des Datensatz, d.h. alles verschiedene Blöcke

- **Bonus (wenn noch Zeit/Lust vorhanden ist):**

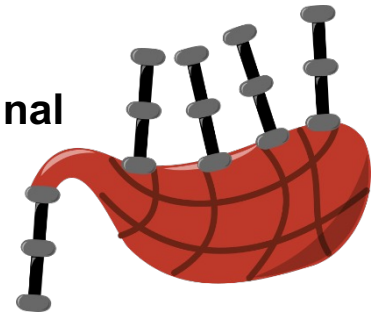
5a	Swingbench	1	100	Stresstest
5b	Swingbench	2	50	Stresstest
5c	Swingbench	3	~33	Stresstest
5c	Swingbench	4	25	Stresstest
6a	Swingbench	1	100/300	OE
6b	Swingbench	2	50/150	OE
6c	Swingbench	3	~33/100	OE
6c	Swingbench	4	25/75	OE

- **Ansatz:**
  - **viele Testfälle**
  - **sicherlich Fehlversuche**
  - **brauchen Reproduzierbarkeit**
  - **also Automatisieren**

- **Starte auf der VM:**



- **Named Pipes:**
  - pro Knoten 4 Linux named pipes (= FIFO) starten
  - was ins FIFO kommt, wird von `tail` herausgezogen und wandert in `SQLPLUS` Terminal
  - `SQLPLUS` mit Option „`-s`“ (kein Banner, Prompt, Command)
  - `SQLPLUS` – Sessions werden an die Knoten „gepinnt“ (kein SCAN)

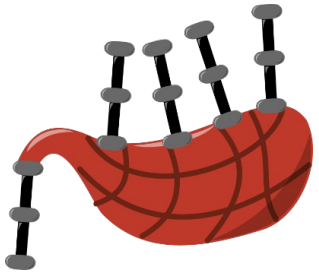


- **Syntax:**
  - einmalig pro Pipe: `mkfifo fifo1-1`
  - pro Testfall, starten der FIFO‘s mit `SQLPLUS` – „Abnehmern“
  - Auszug aus Skript `open-fifos.sh`:

```
tail -f fifo1-1 | sqlplus -s benchmark/bench@10.255.255.12:1521/PDB.Client >
./fifo-1-1.log > /dev/null &
```

- **Drums:**
  - Dampf auf die Maschine
  - pro Pipe ein Prozess
  - trommelt die Pipe mit SQL-Statements zu
  - Skript mit Parametern für Anzahl Knoten und Sessions
  - pro Testfall ein Bash – Skript
  - SQLPLUS arbeitet Befehle so schnell ab, wie es die DB zulässt
  - Zeitmessung in SQL, denn Bash misst nur das Befüllen der Pipe
  - Zeitmessung pro Statement (`SET TIMING ON`)
  - Keine unnötigen Ausgaben (`SET FEEDBACK OFF`)

- **Somit:**



AND

- **Auszug aus Skript `case2.sh`:**

Schleife über die Knoten

```
for (( nodes=1; nodes<=$1; nodes++ )) do
```

Schleife über die Sessions

```
    for (( sessions=1; sessions <=$2; sessions++ )) do
        echo "Node: $nodes Session: $sessions"
```

Endlos – Schleife im Hintergrund

```
        while sleep .06 do
```

STRG+C zulassen

```
            cat > fifo${nodes}-${sessions} <<- EOF
```

Füttern der PIPE

```
                SET TIMING ON;
                SET FEEDBACK OFF;
                UPDATE smallbench SET TEXT='aaa' WHERE ID <= 10000;
                COMMIT;
            EOF
            echo -n "."
        done &
    done;
done;
```

Input für SQLPLUS

Keep-Alive auf Console

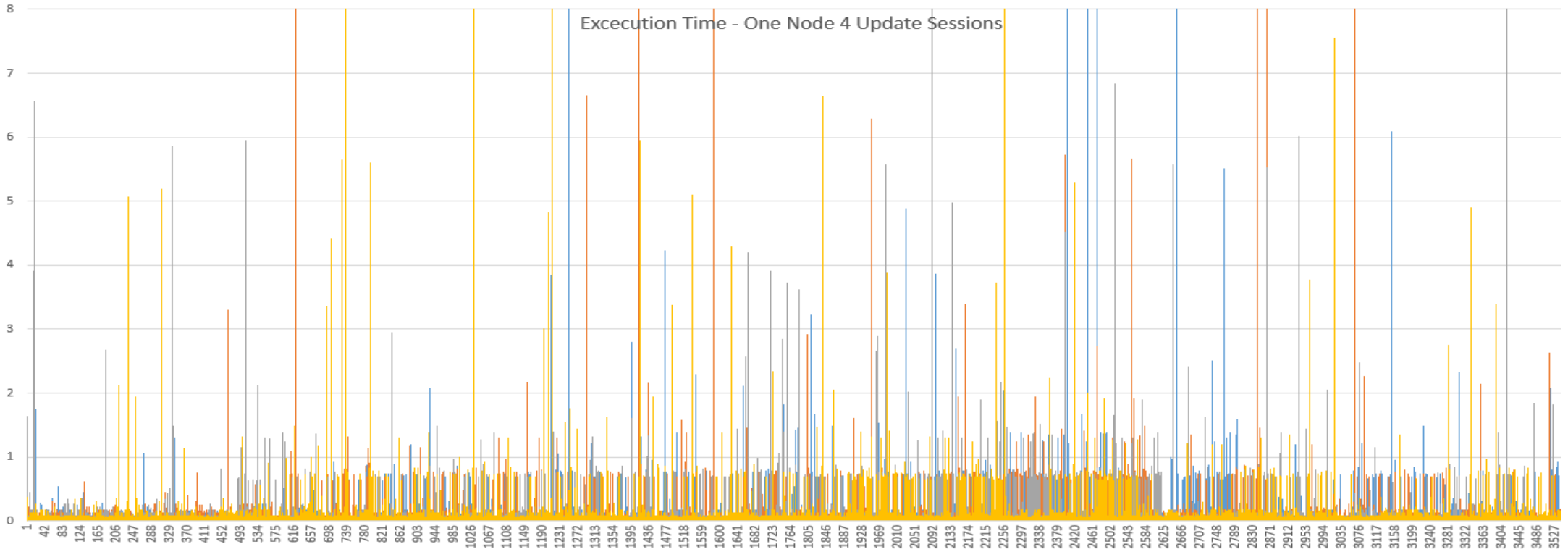
- **Logfile(s) von SQLPLUS:**

```
Abgelaufen: 00:00:00.01
Abgelaufen: 00:00:01.35
Abgelaufen: 00:00:00.02
Abgelaufen: 00:00:01.81
…
```

umformer-awk.sh →

```
00,01
01,35
00,02
01,81
…
```

→ 



- **Rohdaten in Excel:**

Session 1	Session 2	Session 3	Session 4
0,24	0,19	1,64	0,38
0	0	0	0
0,09	0,09	0,09	0,08
0	0	0	0
0,09	0,23	0,12	0,14
0	0	0,01	0
0,13	0,13	0,45	0,13
0	0	0	0
0,14	0,14	0,18	0,14
0	0	0	0
0,13	0,13	0,18	0,13
0	0	0	0
0,14	0,18	0,18	0,13
0	0	0,03	0,01

UPDATE

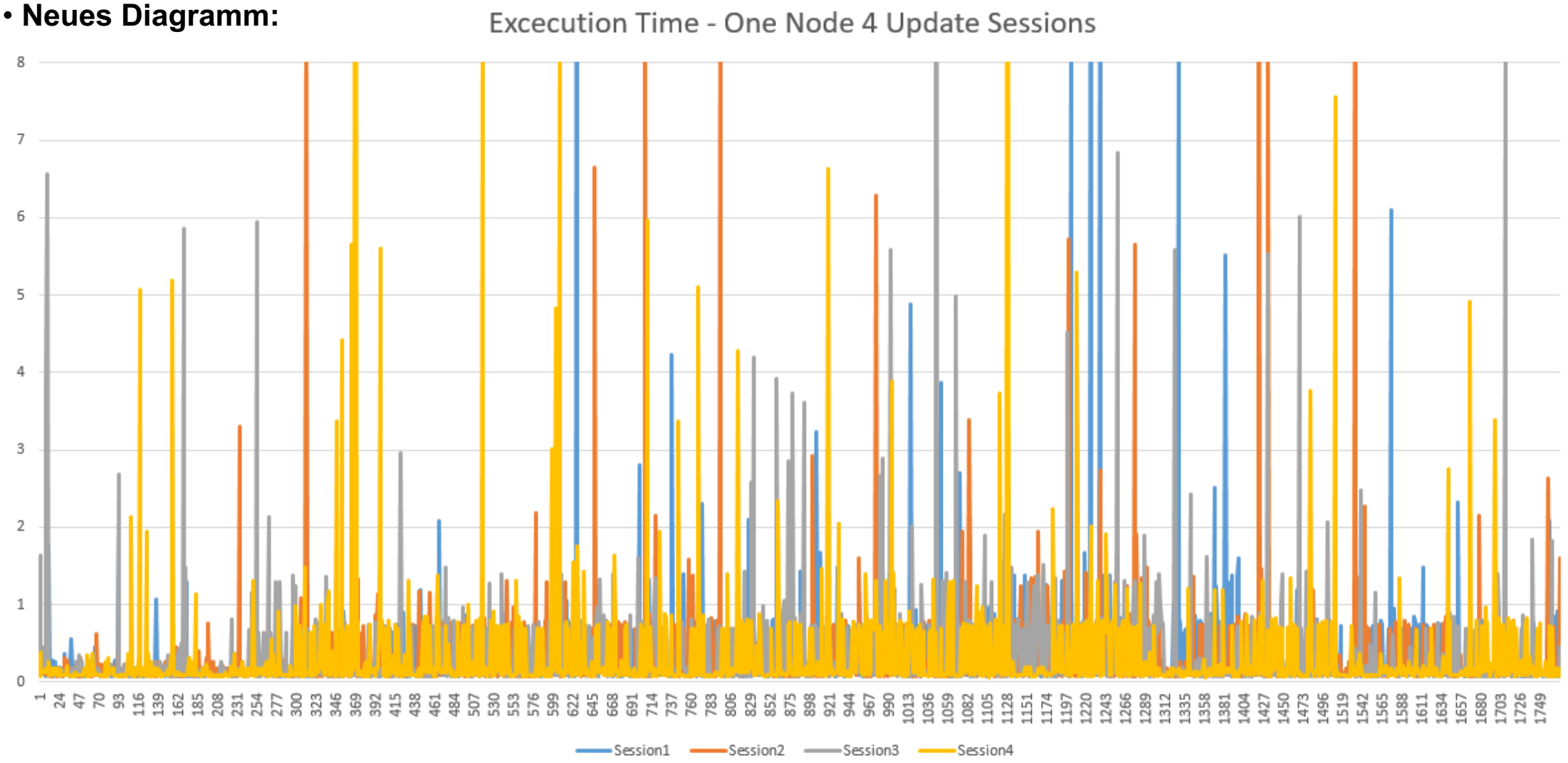
COMMIT

Ergänze Dummy – Spalte (0/1) und filtern:

Session 1	Session 2	Session 3	Session 4	Dummy
0,24	0,19	1,64	0,38	1
0	0	0	0	0
0,09	0,09	0,09	0,08	1
0	0	0	0	0
0,09	0,23	0,12	0,14	1
0	0	0,01	0	0
0,13	0,13	0,45	0,13	1
0	0	0	0	0
0,14	0,14	0,18	0,14	1
0	0	0	0	0
0,13	0,13	0,18	0,13	1
0	0	0	0	0
0,14	0,18	0,18	0,13	1

Excel-Tipp für die Dummy-Spalte mit Formel: REST(ZEILE();2)

- **Neues Diagramm:**



- **Werte mit `COMMIT`:**

	Session 1	Session 2	Session 3	Session 4
Max	25,07	19,23	25,25	17,85
Min	0	0	0	0
Mittelwert	0,16	0,16	0,17	0,17
Median	0,07	0,06	0,05	0,07

- **Werte ohne `COMMIT`:**

	Session 1	Session 2	Session 3	Session 4
Max	25,07	19,23	25,25	17,85
Min	0,07	0,07	0,07	0,06
Mittelwert	0,31	0,31	0,33	0,33
Median	0,13	0,13	0,13	0,13

Knapp verdoppelt!

- Nach Studium von Excel und AWR – Reports:
  - es fehlt offensichtlich noch etwas
  - bisher nur Ausführungszeit pro Statement
  - unterschiedlich viele Statements pro Knoten
  - neu → Wait – Events und Veränderungen
  - Gezielte Untersuchung: `GC current block busy` sowie `enq: TX – row lock contention`

- Idee:
  - wir wollen Operationen = Statements pro Sekunde sehen
  - Daten aus `gv$system_event` und `gv$sqlarea` abziehen
  - Insert dieser Daten in eine Tabelle
  - Scheduler mit `PL/SQL` zum Daten sammeln
  - Daten im Nachgang auswerten

- **Tabelle:**

```
SQL> DESC BENCHMARK.EXEC_PER_TIME
Name                    Type
-----------------------------------
EXEC_PER_TIME_ID        NUMBER        -- Sequence
TIMESTAMPS              TIMESTAMP(6)  -- Zeit
INST_ID                 NUMBER        -- Knoten
SQL_ID                  VARCHAR2(13)  -- rein informativ (PoC)
EXECUTIONS              NUMBER        -- Anzahl zum gemessenen Zeitpunkt
ENQ_TX_ROW_LOCK_CONT    NUMBER        -- Anzahl zum gemessenen Zeitpunkt
GC_CURRENT_BLOCK_BUSY   NUMBER        -- Anzahl zum gemessenen Zeitpunkt
SQL_TEXT                VARCHAR2(100)-- rein informativ (PoC)
```

- **Schedule:**

```
sys.dbms_scheduler.create_job(
    job_name        => 'BENCHMARK.PUT_TO_TABLE',
    job_type        => 'PLSQL_BLOCK',
    job_action      => 'BEGIN BENCHMARK.FILL_TABLE; END;',
    start_date      => TO_TIMESTAMP_TZ('15-08-2022 19:25:00 EUROPE/BERLIN',,DD-MM-YYYY HH24:MI:SS TZR'),
    repeat_interval => 'Freq=Secondly;Interval=1',
    …)
```

- **Auszug aus PL/SQL – Prozedur Benchmark.Fill_Table :**

```
FOR rec IN (

  SELECT SYSTIMESTAMP AS timestamps,
         a.inst_id,
         a.sql_id,
         a.executions,
         a.sql_text,
         (SELECT total_waits
            FROM gv$system_event e
           WHERE e.event_id = 310662678 AND a.inst_id = e.inst_id)  AS enq_tx_row_lock_cont,
         (SELECT total_waits
            FROM gv$system_event e
           WHERE e.event_id = 2701629120 AND a.inst_id = e.inst_id) AS gc_current_block_busy
    FROM gv$sqlarea a
   WHERE a.sql_fulltext LIKE 'update smallbench%
   ORDER BY a.inst_id, a.sql_id )

LOOP
  INSERT INTO benchmark.exec_per_time (exec_per_time_id, timestamps, inst_id, executions, …);
END LOOP;
```

- **Auszug Rohdaten Fall 4b (4 Knoten a 1 Session mit Update auf 1 Datensatz):**

ZEITPUNKT	INST_ID	EXECUTIONS	ENQ_TX_ROW_LOCK_CONT	GC_CURRENT_BLOCK_BUSY	SQL_TEXT
26-08-2022 15:10:41	1	172		218	update smallbench set TEXT='aaa' where ID=1*10000
26-08-2022 15:10:41	2	186		106	update smallbench set TEXT='aaa' where ID=2*10000
26-08-2022 15:10:41	3	179		129	update smallbench set TEXT='aaa' where ID=3*10000
26-08-2022 15:10:41	4	175		218	update smallbench set TEXT='aaa' where ID=4*10000
26-08-2022 15:10:42	1	325		218	update smallbench set TEXT='aaa' where ID=1*10000
26-08-2022 15:10:42	2	342		106	update smallbench set TEXT='aaa' where ID=2*10000
26-08-2022 15:10:42	3	329		129	update smallbench set TEXT='aaa' where ID=3*10000
26-08-2022 15:10:42	4	326		218	update smallbench set TEXT='aaa' where ID=4*10000

- **Mit SQL-Funktion (LAG) die Differenz zum vorherigen Datensatz ermitteln:**

ZEITPUNKT	INST_ID	DIFF_EXEC	DIFF_ENQ_TX_ROW_LOCK_CONT	DIFF_GC_CURRENT_BLOCK_BUSY
26-08-2022 15:10:42	1	153	0	0
26-08-2022 15:10:42	2	156	0	0
26-08-2022 15:10:42	3	150	0	0
26-08-2022 15:10:42	4	151	0	0
26-08-2022 15:10:43	1	100	0	0
26-08-2022 15:10:43	2	102	0	0
26-08-2022 15:10:43	3	106	0	0
26-08-2022 15:10:43	4	97	0	0

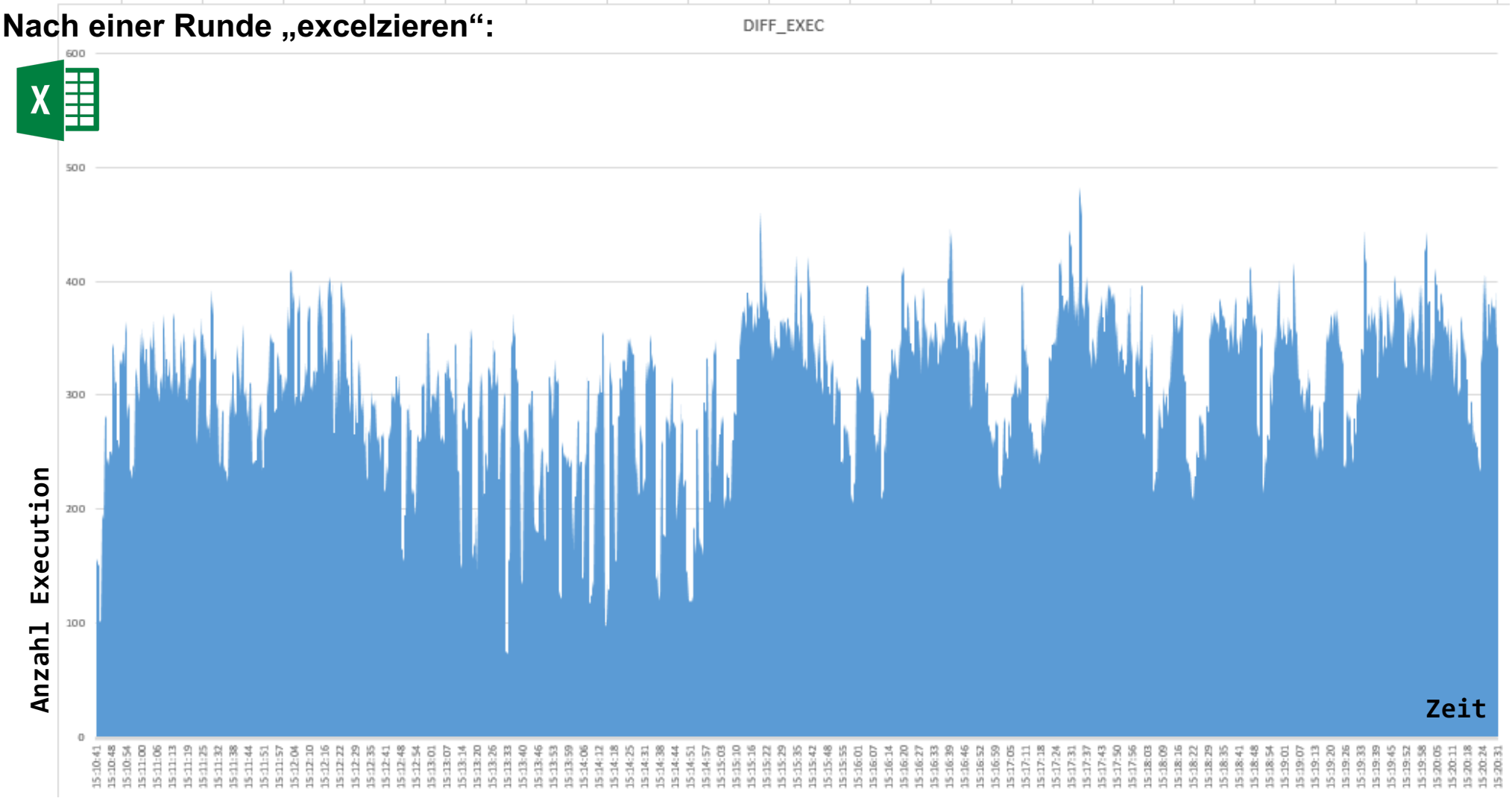
153 = 325 – 172

0 = 218 – 218

0 erwartet, da verschiedene Datensätze

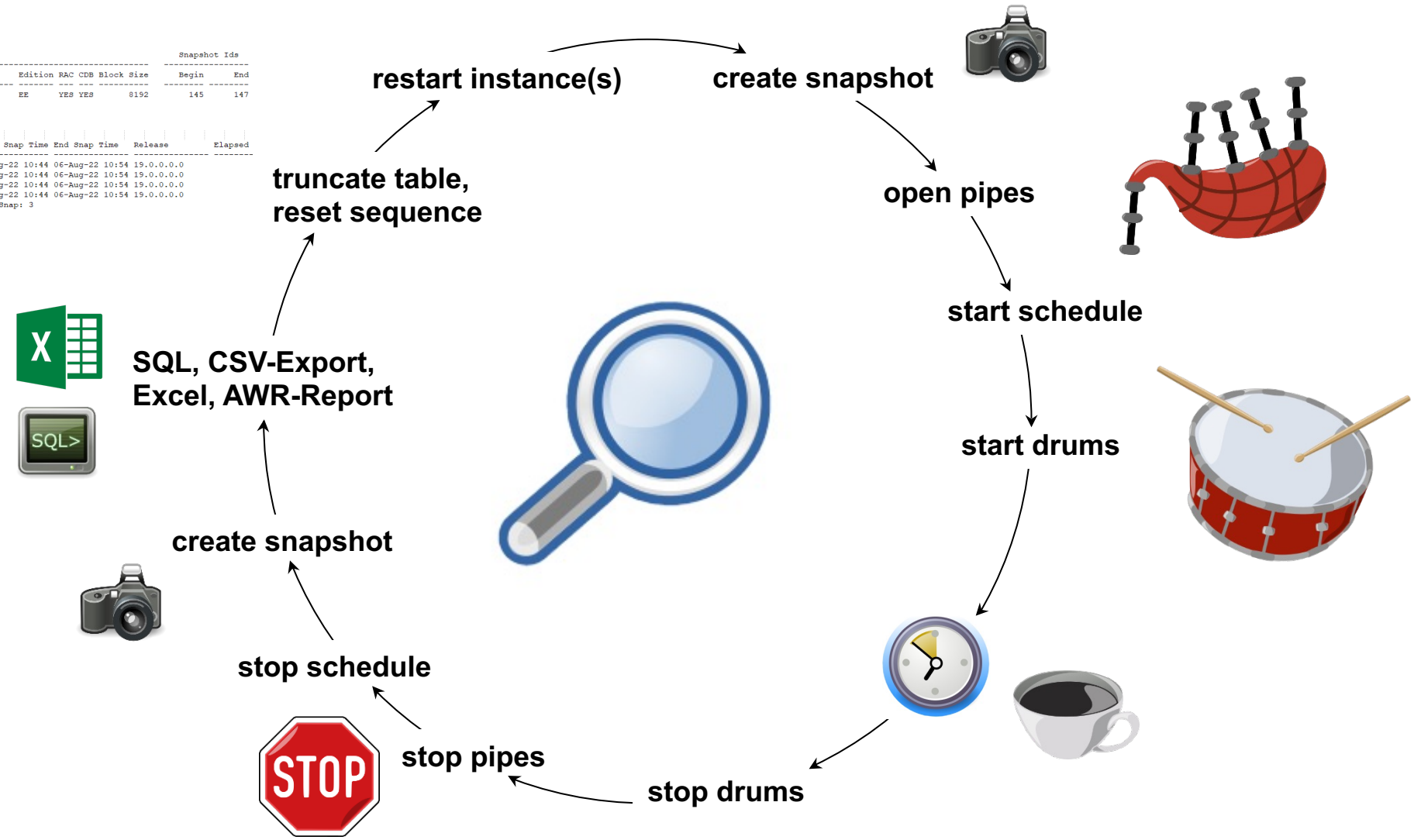
0 erwartet, da verschiedene Blöcke

- **Nach einer Runde „excelzieren“:**

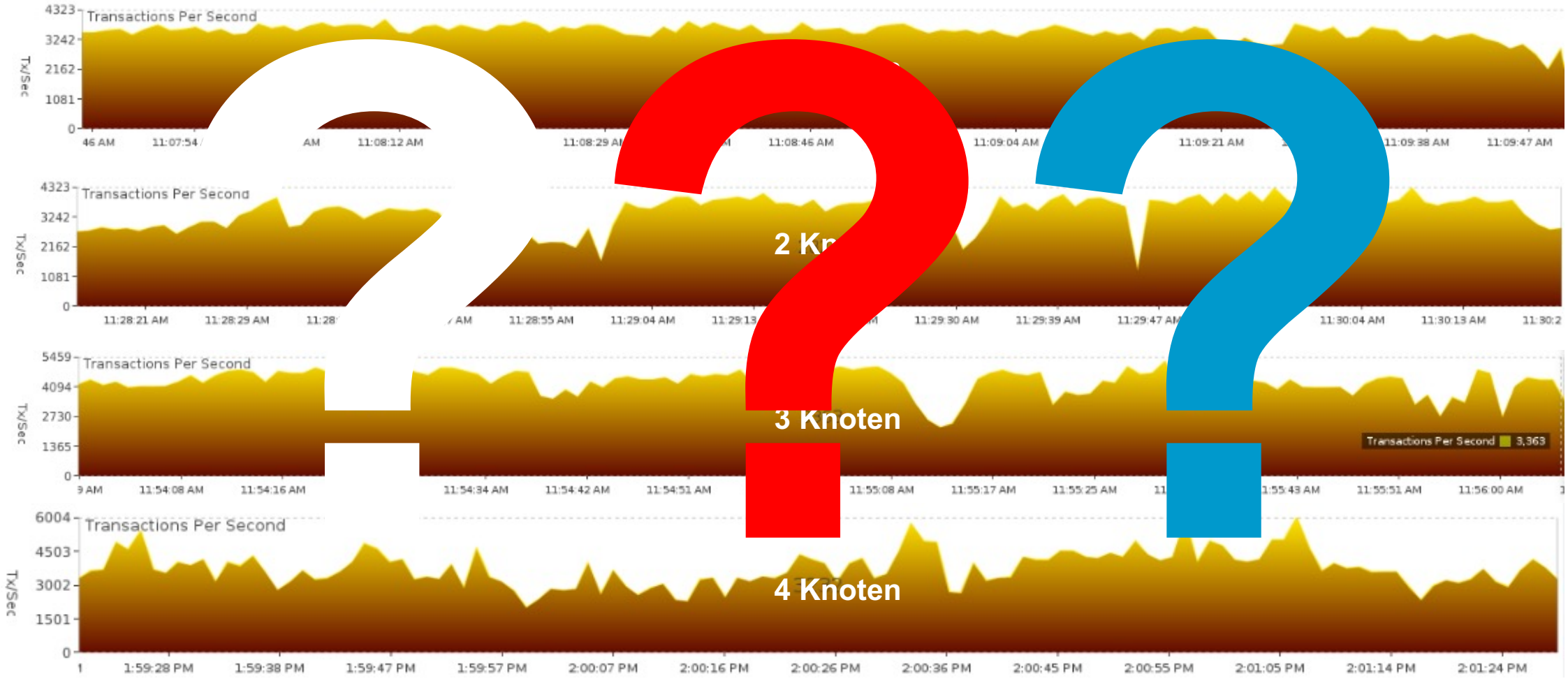


Tools: Kreis des Wahnsinns / Kreislauf der Testfälle

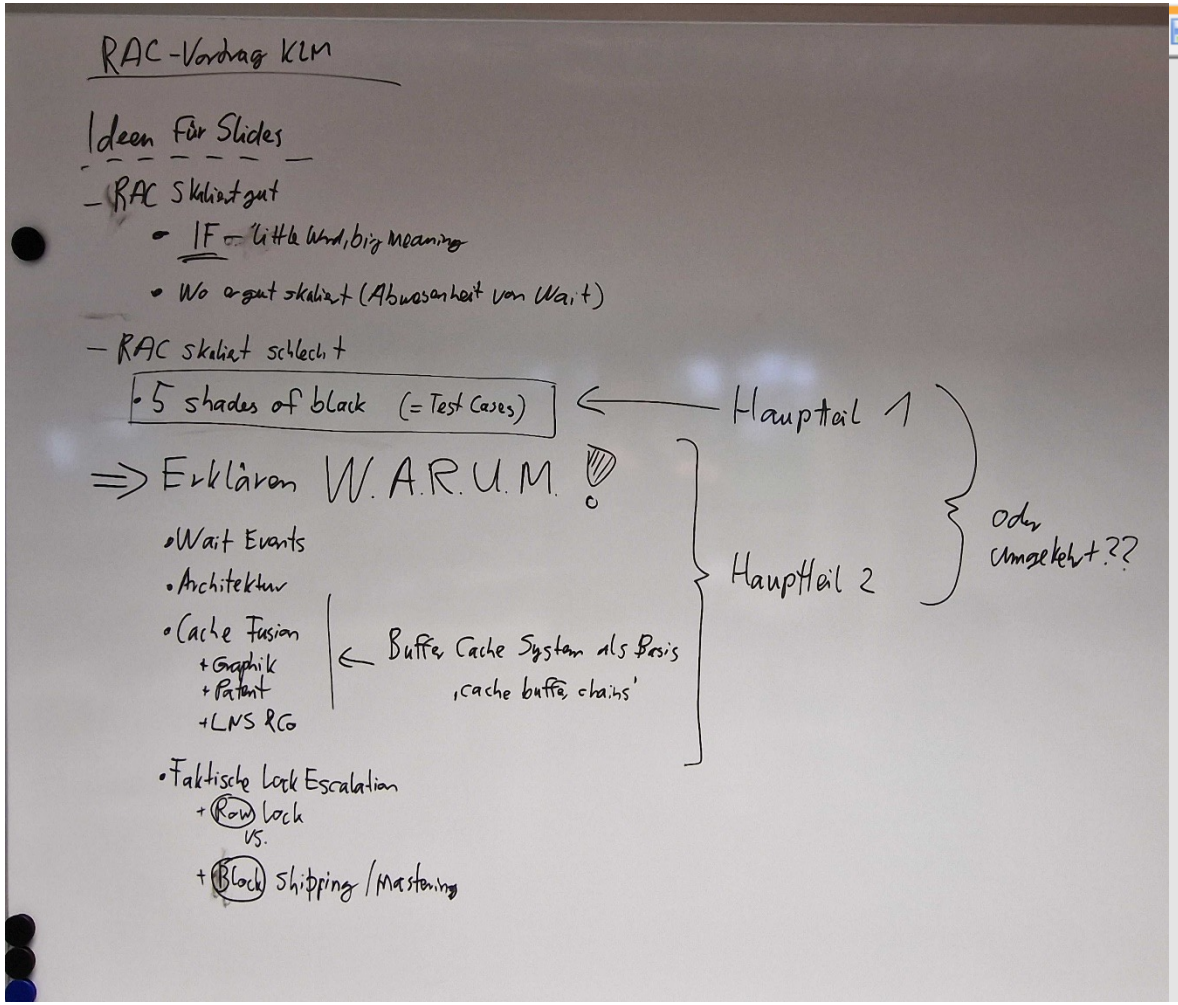
```
WORKLOAD REPOSITORY REPORT (RAC)

Database Summary
~~~~~~~~~~~~~~~~
                                Database                                    Snapshot Ids
---------------------------------------------------------------------------  ---------------
       Id Name      Unique Name Role            Edition RAC CDB Block Size     Begin       End
---------- --------- ----------- --------------- ------- --- --- ----------  -------- --------
1250448516 BSSGT     BSSGTA      PRIMARY         EE      YES YES        8192       145      147
Database Instances Included In Report
-> Listed in order of instance number, I#
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  I# Instance  Host        Startup       Begin Snap Time End Snap Time   Release         Elapsed
---- --------- ----------- ------------- --------------- --------------- --------------- ---------
   1 BSSGTA_1  bss-inhous 06-Aug-22 10:42 06-Aug-22 10:44 06-Aug-22 10:54 19.0.0.0.0
   2 BSSGTA_2  bss-inhous 06-Aug-22 10:42 06-Aug-22 10:44 06-Aug-22 10:54 19.0.0.0.0
   3 BSSGTA_3  bss-inhous 06-Aug-22 10:42 06-Aug-22 10:44 06-Aug-22 10:54 19.0.0.0.0
   4 BSSGTA_4  bss-inhous 06-Aug-22 10:42 06-Aug-22 10:44 06-Aug-22 10:54 19.0.0.0.0
Open Pluggable Databases at Begin Snap: 3, End Snap: 3
~~~~~~~~~~~~~~~~~~~~~~~~~
```



- Da war doch noch was … Swingbench Order Entry (OE) Benchmark:



Tools: Foliendesign …



Roter-Faden-Making-Of-Short.txt

```
1) Motivation
      - Lesson Learned vom Data Guard Vortrag

2) Herausforderungen

   Zusammenarbeit
   Das Labor
   Überlegung was wollen wir wie zeigen?
   - Testfälle CASE 1 - 4 plus Bonus?

3) Tools - Technik & Auswertung:

    Die Idee erklären
     - Bash, SQL und PL/SQL
         ==> Pipe & Drums
         ==> Nächster Datensammler
         1.) Cluster-Event:     GC current block busy
         2.) Application-Event: enq: TX - row lock contention

     - Swingbench ???

     - Kreislauf des Wahnsinns

4) Zusammenfassung

   Outtakes
     - "Sag mal Leute, geht Ihr mir heute auf den Sack ..."
     - "Würde es zeigen wollen, allein schon weil wir es können."
```

Tools: Foliendesign …

Agenda …

1.) Motivation

2.) Herausforderungen

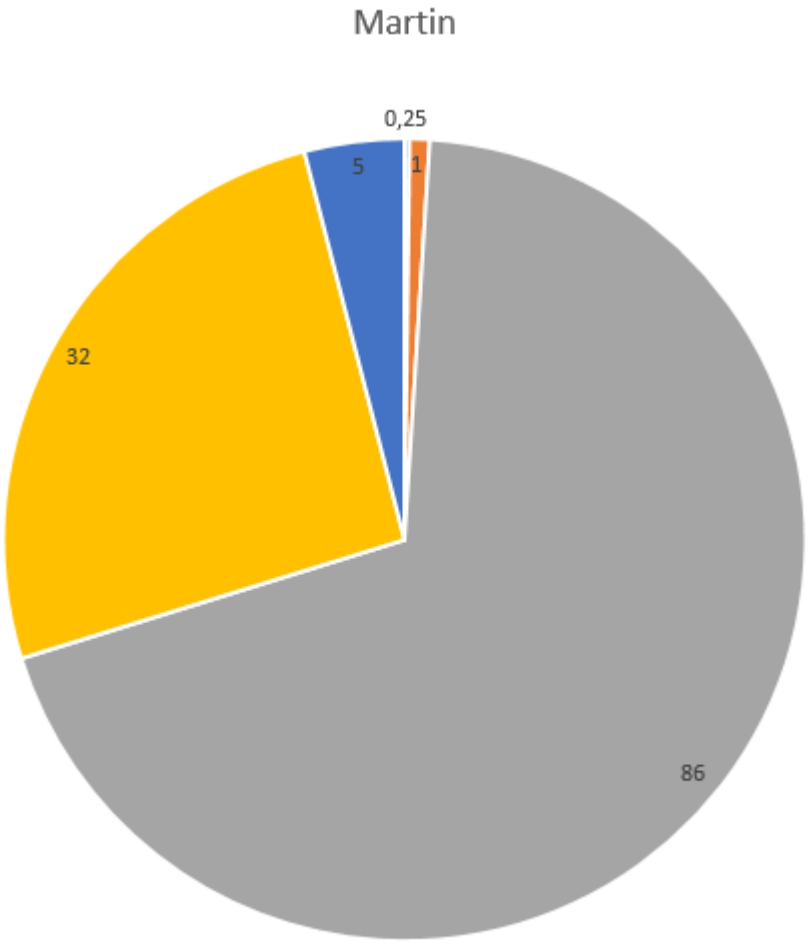
3.) Tools

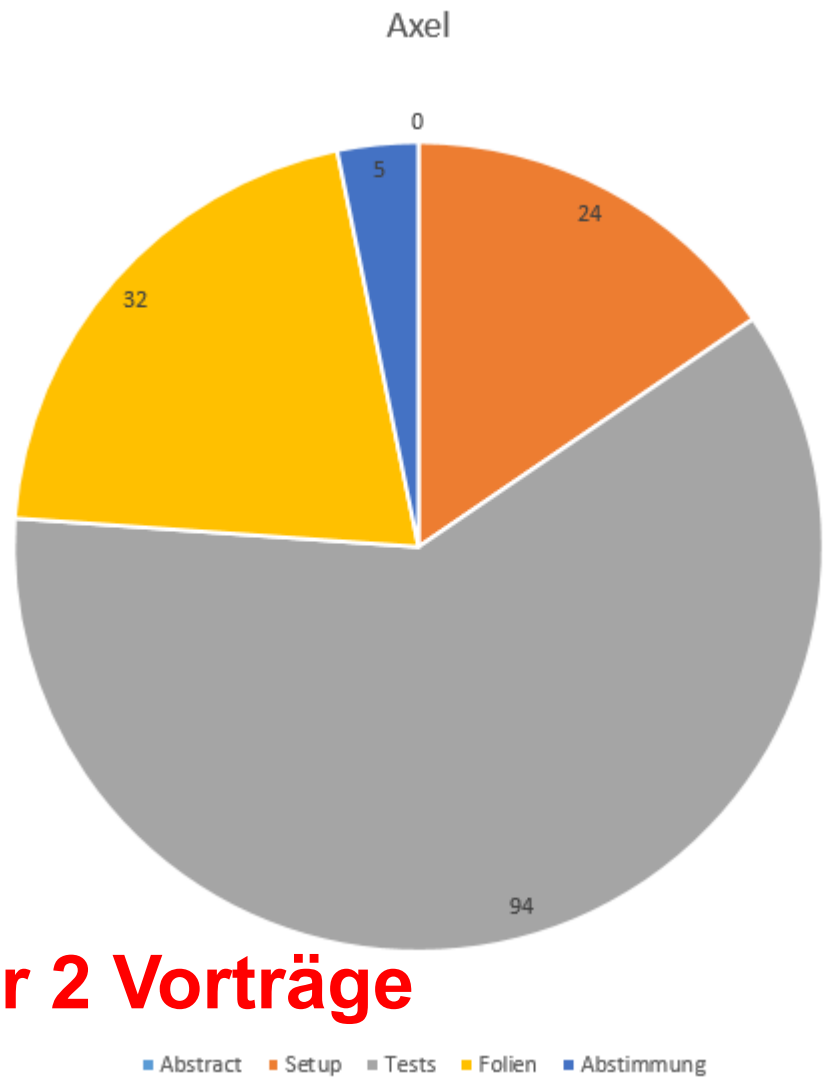
4.) Zusammenfassung

Zusammenfassung: Fazit

- es hat wieder sehr viel Spaß gemacht
- diese Art von Vortrag kostet richtig viel Zeit
- es bestätigt sich wieder: traue keiner Statistik, die Du nicht selbst gefälscht hast
- zwei Vorträge zu gestalten, die inhaltlich zusammengehören ist eine Herausforderung
- RAC einzusetzen ohne sich die Applikation anzusehen, kann nach hinten losgehen
- RAC ist für die aktuelle BSS-Anwendung nur für ganz ausgewählte Teile sinnvoll
- Danke an alle Personen im Hintergrund: Sys-Admin mit VPN, Storage, etc., DBA‘s Installation RAC, etc.

Zusammenfassung: Fazit





ca. 280 Stunden für 2 Vorträge

- „RAC ist wie RAC One Node nur ohne One Node.“
- „Namenskonventionen sind Scheiße - aber Scheiße ist unabdingbar.“

- „Den Hammer ham wa! Sprich wir sind auf der Suche nach den Nägeln ...“

- „Sensation! Durch hinzufügen von Knoten haben wir es geschafft die Leistung zu halbieren. “

- „Jemand, der Dich nervt hat meistens gar keinen Stress damit! “
- „Da müssen wir noch excelzieren ... “

- „Excel ist nervig ... also was für Buchhalter.“
- „Töte nichts was funktioniert.“
- „Folien sind was für‘s Hotel.“

Mein Favorit:

Zusammenfassung: Quellen

- **Oracle Real Application Clusters 19c Technical Architecture:**

  https://www.oracle.com/webfolder/technetwork/tutorials/architecture-diagrams/19/rac/main.html

- **ORION:**

  https://docs.oracle.com/en/database/oracle/oracle-database/19/tgdba/IO-configuration-and-design.html#GUID-355C99D8-29C1-421F-8B65-47A3C48324A2

- **Swingbench:**

  https://www.dominicgiles.com/swingbench.html

  https://www.dominicgiles.com/clusteroverviewwalkthough23.html

  https://www.doag.org/de/home/news/aufgezeichnet-die-oracle-datenbank-mittels-swingbench-besser-verstehen

Was skaliert gut im RAC?

Wenn jeder sein Ding machen kann ...
das ist wie im wahren Leben!

Ein Vortrag muss nicht immer nur Technik in aller Tiefe sein:

Berichten Sie doch auch

mal über IHR „Making of“!

axel.vomstein@bss.gmbh

martin.klier@performing-db.com