

# HERZLICH WILLKOMMEN

## Making of: Der Preis den wir zahlen – Performance und HA mit Data Guard

**Axel vom Stein**

mit Martin Klier  **Performing Databases GmbH**

**Hauptsitz: Solingen (NRW)**

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

**Gegründet: 1991**

**Rechtsform: GmbH**

**seit 2006: Unternehmen der ROFA Industrial Automation Group**

**Mitarbeiter: ca. 160**



- **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.**



- **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/Fables:**

- **Ausbildung**
- **Sicherheits- und HA - Konzepte**
- **Migrationsprojekte**
- **Standardisierung**
- **alles was sich automatisieren lässt**



# 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

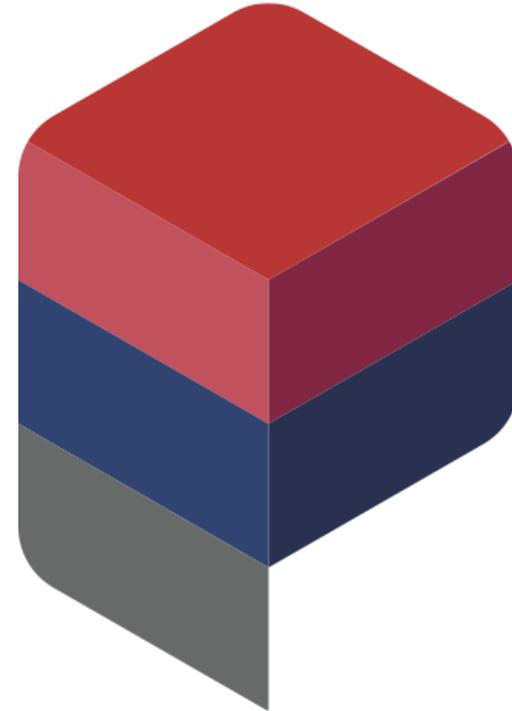


**ORACLE**<sup>®</sup>  
ACE Director



# 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



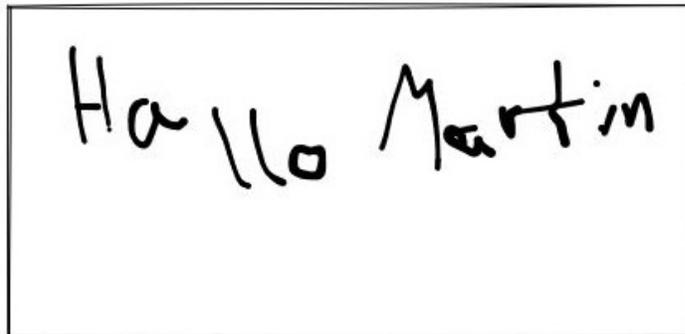
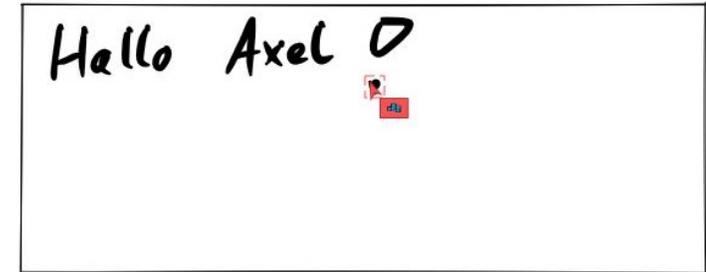
**1.) Motivation**

**2.) Herausforderungen**

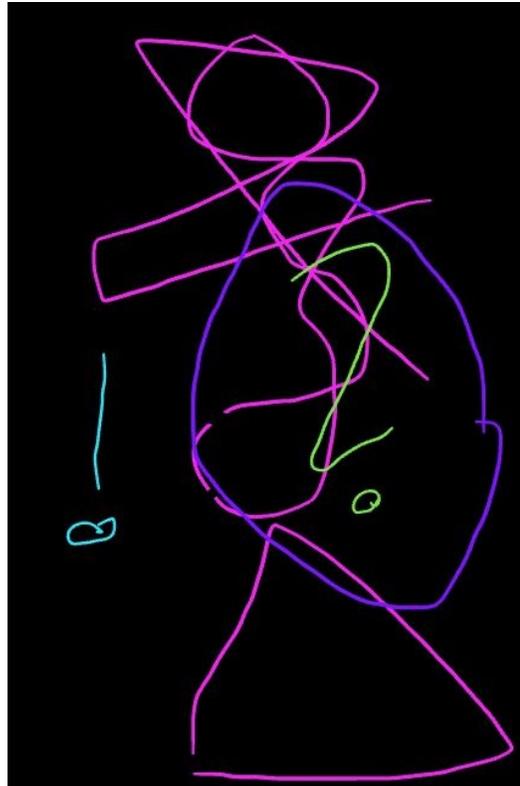
**3.) Tools**

**4.) Zusammenfassung**

Die Hauptdarsteller:



### Ist das Kunst oder kann das weg?



### Unsere Kommunikation:

Martin Klier Gestern, 9:36

 blblbl 😊

Sie Gestern, 9:57

 Danke, wieder was neues gelernt! 🗨️ 😊 ...

Sie 8:30

 Moin

Ist Deine Spielweise offline?

Martin Klier 8:32

 📞 Eingehender Anruf - Abgenommen

Martin Klier 8:32

 Machmer Meeting? 😊

**Die besten Sprüche:**

- **„Unfassbar: Die Dinge, die in der Doku stehen - existieren auch wirklich!“ ← wir werden sehen ...**
- **„Ich mach mal dunkel – ich sehe nämlich nix und das ist kein Widerspruch!“**
- **„Wir ziehen die Messgröße ran, die den größten Showeffekt hat.“**
- **„Mit Brille geht’s – ohne nicht!“**
- **„Wenn Größe zählt, dann können wir nochmal draufdrücken.“**
- **„Wenn das Ergebnis nicht spektakulär ist, fangen wir halt von vorne an.“**
- **„Das ist der rohe Plan, aber der wird in der Praxis nicht standhalten.“**

- **mehrjährige Zusammenarbeit in Projekten des Logistikumfeldes**
  - **Sizing für Infrastruktur bisher per „Daumen“ und Erfahrungswerten**
  - **Wunsch belegbare Zahlen zu erhalten**
  - **Was „klaut“ uns HA (hier Data Guard) eigentlich an Leistung?**
  - **Gefühl/Hoffnung/Erwartungshaltung: der Verlust an Leistung liegt unter 5%**
  - **Recherche im Netz liefert wenig Output**
- also probieren und Erkenntnisse teilen ...**

- Oracle 19c (19.9) Multitenant Architektur mit Data Guard
- ASM mit Diskgroups (DATA, FRA, REDO1, REDO2)
- 2 Oracle Linux VM's (Primary/Standby) auf unterschiedlichen ESXi Hosts
- Testumgebung ist CPU-mäßig sehr stark im Verhältnis zum Storage



1.) Motivation

2.) Herausforderungen

3.) Tools

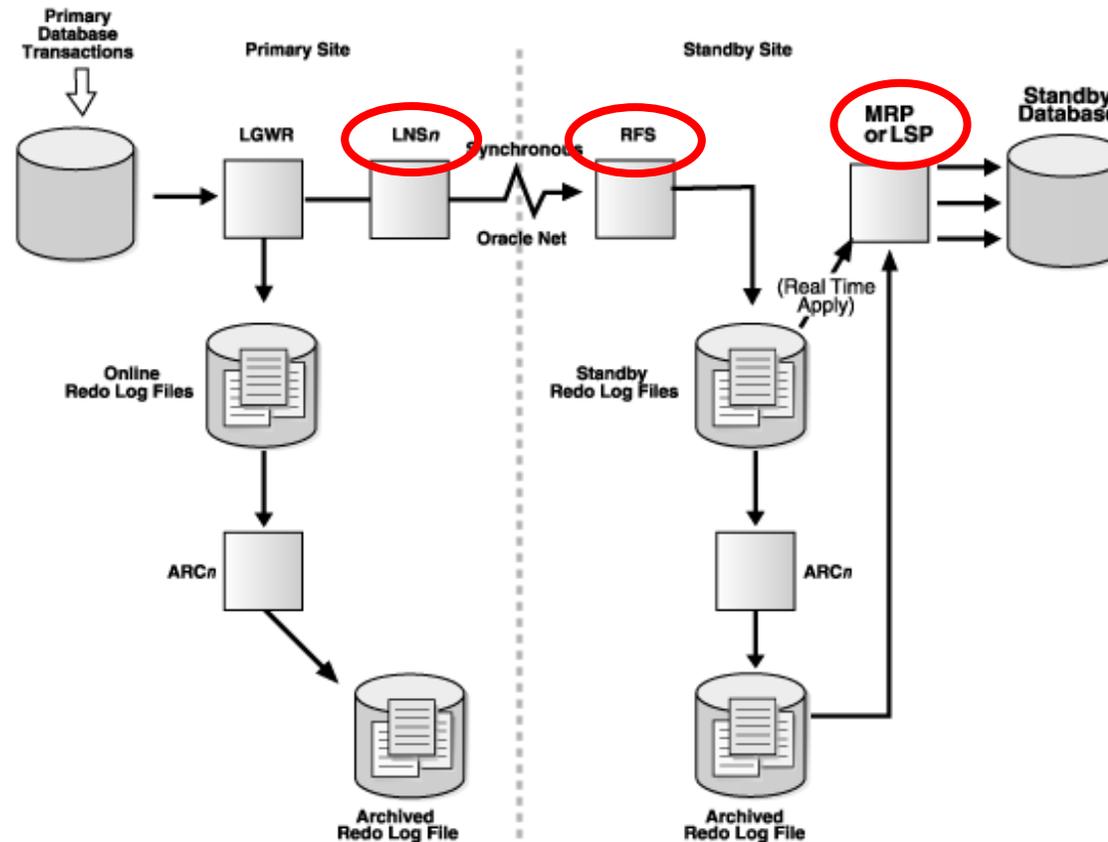
4.) Zusammenfassung





- Oracle Dokumentation bis 10.2 (Kapitel RedoTransport Service)

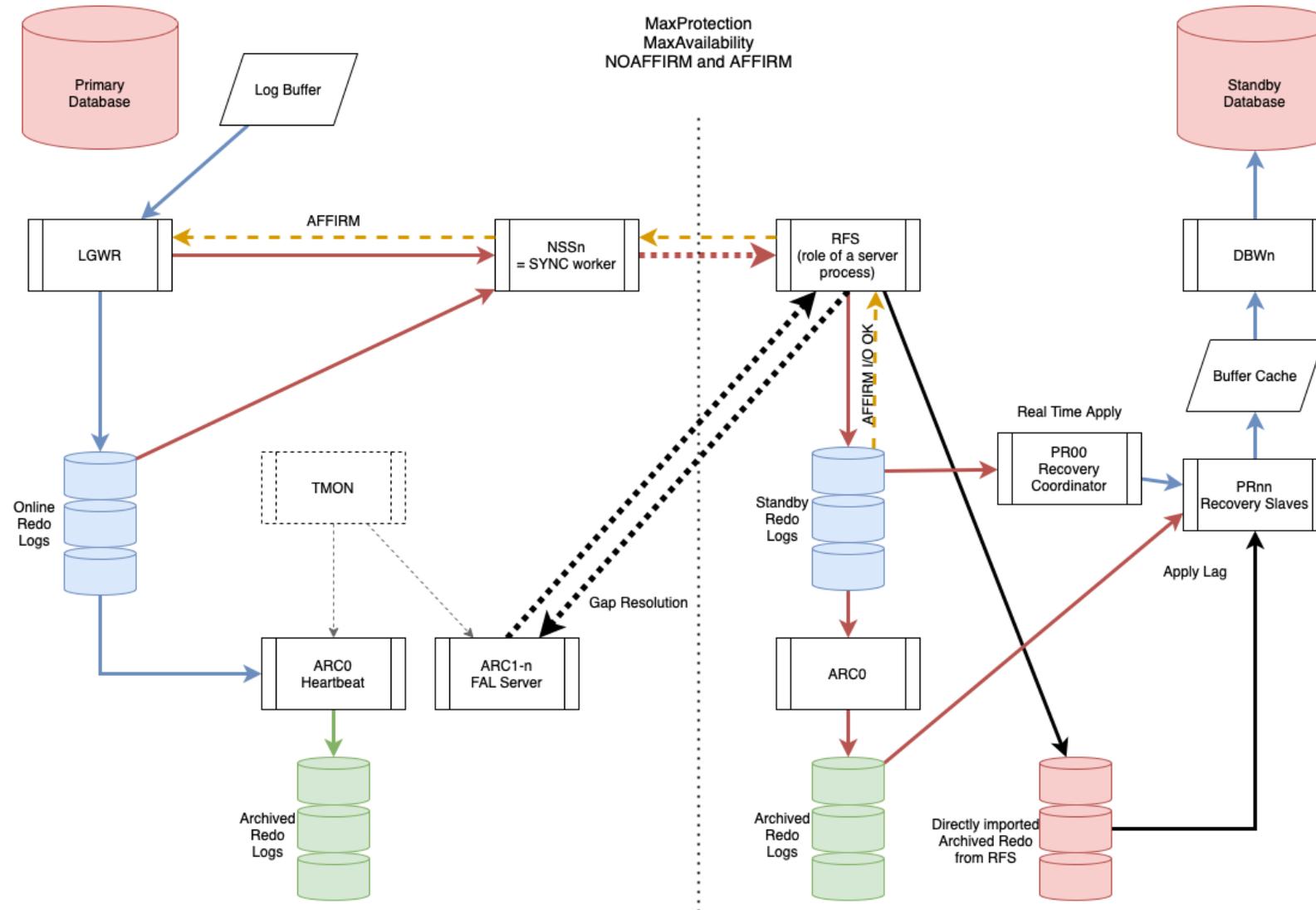
Figure 5-4 LGWR SYNC Archival to a Remote Destination with Standby Redo Log Files



- Oracle Dokumentation ab 11g (Kapitel RedoTransport Service)



# Herausforderungen: Data Guard Architektur Reloaded



- Wie sind wir auf das Diagramm gekommen (ein Auszug)?

```
ps -ef | grep lns → NIX ??
```

```
SQL> select name,pid,role from v$dataguard_process order by name;
```

NAME	PID	ROLE
ARC0	17356	archive local
ARC1	17364	archive redo
DMON	17137	broker monitor
INSV	17415	broker instance slave
LGWR	17097	log writer
<b>NSS3</b>	<b>25539</b>	<b>sync</b>
NSV0	17424	broker net slave
NSV2	27519	broker net slave
RSM0	17500	broker worker
TMON	17175	redo transport monitor
TT00	17358	gap manager
TT01	17360	redo transport timer
TT02	27535	heartbeat redo informer
TT03	22996	controlfile update

• NSSx = LNSx?

• x = 3?

- **Auf der Primary suchen:**

```
netstat -tpn | grep nss
tcp        0      0 10.160.122.10:34930    10.160.122.34:1534    ESTABLISHED 25539/ora_nss3

ps -ef | grep ora_nss3
oracle    25539      1  0 09:46 ?          00:00:01 ora_nss3_FLL1PA_1
```

- **Mit dem Socket auf der Standby – Seite suchen:**

```
netstat -tpn | grep 10.160.122.10:34930
tcp        0      0 10.160.122.34:1534    10.160.122.10:34930    ESTABLISHED 16475/oracleFLL1PB
```

```
SQL> oradebug setospid 16475;
Oracle pid: 53, Unix process pid: 16475, image: oracle@bss-05-ora3
SQL> oradebug tracefile_name;
/oracle/base/db/diag/rdbms/fl11pb/FLL1PB/trace/FLL1PB_rfs_16475.trc
SQL> oradebug flush;
```

➔ **LNS ist raus**

➔ **NSS & RFS unterhalten sich über Sockets**

➔ **Rolle des RFS wird von ganz normaler Serverprozess auf der Standby-Seite übernommen**

- Hä? – Effekte und zeitliche Aspekte bei Messreihen (später)
- Auszug möglicher Testfälle:

Mode	SYNC	AFFIRM	Standby Redo Logs	Apply Lag	Transport Lag
Baseline - no DG	n/a	n/a	n/a	n/a	n/a
Max Performance	ASync	NOAFFIRM	nein	ja	nein
Max Performance	ASync	NOAFFIRM	ja	nein	ja
<del>Max Performance</del>	<del>ASync</del>	<del>NOAFFIRM</del>	<del>nein</del>	<del>nein</del>	<del>nein</del>
<del>Max Performance</del>	<del>ASync</del>	<del>NOAFFIRM</del>	<del>ja</del>	<del>nein</del>	<del>ja</del>
<del>Max Performance</del>	<del>ASync</del>	<del>NOAFFIRM</del>	<del>ja</del>	<del>nein</del>	<del>nein</del>
MaxAvailability	SYNC	NOAFFIRM	ja	n/a	n/a
MaxAvailability	SYNC	AFFIRM	ja	n/a	n/a
<del>Max Protection</del>	<del>SYNC</del>	<del>AFFIRM</del>	<del>ja</del>	<del>n/a</del>	<del>n/a</del>

→ somit bleiben vier Fälle plus die Baseline

**1.) Motivation**

**2.) Herausforderungen**

**3.) Tools**

**4.) Zusammenfassung**

### Mix aus Lastverursacher und graphischer Darstellung:

- **dd**
- **ORION**
- **Excel**
- **Swingbench**
- **AWR snapshots**
- **nmon**
- **nmonchart**

- eigenes Volume zum Test der Grundlast (lesend & schreibend)
- Nutzung von dd, da ORION (später) kein Multiblock-Read kann
- Test mit dev/zero und dev/urandom, Wert 1024k da DB\_FILE\_MULTIBLOCK\_READ\_COUNT=128

```
[oracle@bench1]$ dd if=/dev/zero of=/dev/BENCHMARK1 bs=1024k count=0
10000+0 records in1000
10000+0 records out
10485760000 bytes (10 GB, 9.8 GiB) copied, 23.9062 s, 479 MB/s
```

```
[oracle@obench1]$ dd if=/dev/urandom of=/dev/BENCHMARK1 bs=1024k count=10000
10000+0 records in
10000+0 records out
10485760000 bytes (10 GB, 9.8 GiB) copied, 238.682 s, 43.9 MB/s
```

- Erkenntnis: Storage ist „blöd“ und ein dev/zero reicht aus, zweite Testreihe mit 8k liefert ebenfalls:
- schreibend ~ 450 – 500 MB/s und lesend ~ 200 – 250 MB/s

- Tool zum Erzeugen von I/O-Last mit gleichem Software Stack wie die DB selbst
- Teil der Oracle-Software (DB/GI), Nutzung auch ohne DB: `$ORACLE_HOME/bin/orion`
- diverse Lastcharakteristika möglich OLTP/DWH sowie Mix
- mit den Daten kann eine Matrix-Grafik mit z.B. Excel erstellt werden
- benötigt wird: Testname, Device (Datei geht auch), Shell-Skript, lun-File
- Skript: 

```
$ORACLE_HOME/bin/orion -run advanced -testname 8disk_4small_16GBcache_80write \  
-num_disks 8 -size_small 4 -type rand -simulate raid0 \  
-write 80 -cache_size 16384 -duration 30 -matrix basic \  
-rtstats_interval 30
```
- lun-File: `/dev/BENCHMARK1`

```
ORION: ORacle IO Numbers -- Version RDBMS_19.3.0.0.ODB RU_LINUX.X64_190417
8disk_4small_16GBcache_80write_20220524_2045
Calibration will take approximately 20 minutes.
Using a large value for -cache_size may take longer.
```

Interval (s)	Num-Read-IOs	Avg-Read-Lat (us)	(Std.Dev )	Read-IOPS (MBps)	Num-Write-IOs	Avg-Wr-Lat (us)	(Std.Dev )	Write-IOPS (MBps)
30.05	1910	13338.60	(6687.82 )	0.25	7404	605.86	(197.46 )	0.96
30.00	3030	17543.80	(13798.08 )	0.39	12112	560.12	(322.88 )	1.58
30.01	3829	28987.41	(26613.49 )	0.50	15444	563.37	(454.10 )	2.01
30.00	4323	39244.37	(33862.33 )	0.56	17333	551.00	(255.63 )	2.26
30.03	4914	46478.90	(44978.76 )	0.64	19714	531.62	(326.64 )	2.56
30.01	5639	50476.91	(38759.63 )	0.73	22318	640.35	(1918.90 )	2.91

- Messreihen mit 4k, 8k, 16k durchgeführt

- Dateien pro Test: 8disk\_4small\_16GBcache\_80write\_20220524\_2045\_trace.txt  
8disk\_4small\_16GBcache\_80write\_20220524\_2045\_mbps.csv  
8disk\_4small\_16GBcache\_80write\_20220524\_2045\_lat.csv  
**8disk\_4small\_16GBcache\_80write\_20220524\_2045\_iops.csv**  
8disk\_4small\_16GBcache\_80write\_20220524\_2045\_hist.txt  
8disk\_4small\_16GBcache\_80write\_20220524\_2045\_summary.txt

- Auszug aus \*iops.csv:

```
Large/Small, 1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, ...
              0, 398, 782, 1091, 1262, 1427, 1747, 1971, 2263, 2483, 2596, 2657, ...
```

# Tools: Excel

**1** Daten

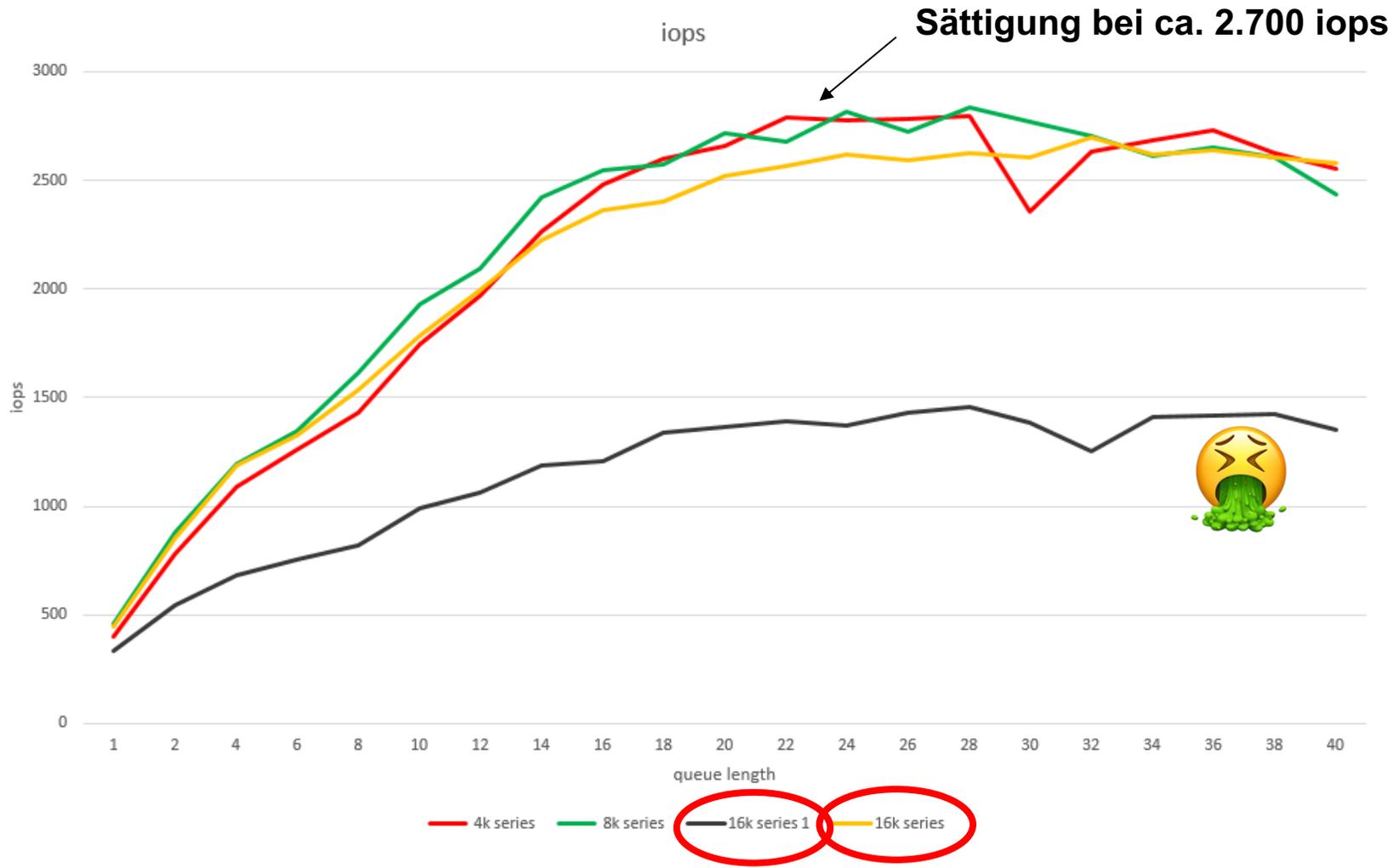
**2** Wählen Sie den Dateityp, der Ihre Daten am besten beschreibt:  
 Getrennt - Zeichen wie z.B. Kommas oder Tabstopps trennen Felder (Excel 4.0-Standard).  
 Feste Breite - Felder sind in Spalten ausgerichtet, mit Leerzeichen zwischen jedem Feld.

**3** Trennzeichen  
 Tabstopp  
 Semikolon  
 Komma  
 Leerzeichen  
 Andere:

**4** Einfügen

**5** Achsenbeschriftungen  
Achsenbeschriftungsbereich: =Tabelle4!\$B\$7:\$V\$7 = 1; 2; 4; 6; 8;...  
OK Abbrechen

**6** Datenreihen formatieren  
Datenreihenoptionen  
Linie Markierung  
Keine Linie  
Einfarbige Linie  
Farbverlaufslinie  
Automatisch



- freier Lastgenerator um Oracle Datenbanken zu belasten (Dominic Giles), aktuell Version 2.6.x
- diverse Benchmarks, wir haben Order Entry (OE) und Stresstest genutzt
- Order Entry (inhomogen, wechselnde Last, gut für DB-Benchmarks)
  - basiert auf OE Beispiel Schema von Oracle
  - einmalige Generierung der Daten durch OE-Wizard
  - Erzeugung eines Schemas mit diversen Tabellen (Parent/Child) und Daten
  - kommt BSS-Applikation nahe
- Stresstest (homogen, gut für Infrastruktur-Benchmarks)
  - I, U, D, S auf eine einzelne Tabelle
  - man kann sofort loslegen, keine Generierung von Daten im Vorfeld notwendig

- Last wird durch ausgeführten PL/SQL-Code erzeugt
- nicht nur Lastgenerator, sondern auch grafische Darstellung
- Kommandozeile (Charbench) oder grafische Oberfläche (Minibench, Swingbench)
- Nutzung von config-files (xml) für die verschiedenen Benchmark-Parametrierungen
- Swingbench kann auch AWR-Snapshots ziehen
- Installation
  - Java 8+ notwendig
  - unzip und einfach starten `./oewizard` bzw. `./swingbench`

Oracle Entry Install Wizard : 2.6.0.1137

File Log Viewer

**Welcome to the**

**Sizing Details**

This wizard will walk you through the process of creating a benchmark. You will need to select a tablespace, users, tables, and indexes.

Select one of the preconfigured sizes for the benchmark. Scaling factor of 1 = 1GB. Based on the sizing details, the schema size is 42.1 GB for a CPU Intensive workload for a more I/O intensive workload.

1 GB  
 10 GB  
 100 GB  
 1 TB  
 User Defined Scale

Author : Dominic Giles  
Email : dominic.giles@

OrderEntry tablespace size =  
Temporary tablespace size required =

Oracle Entry Install Wizard : 2.6.0.1137

File Log Viewer

**"Building the Simple Order Entry" Schema**

Completed schema successfully

Statistic	Value
Connection Time	0:00:00.003
Data Generation Time	0:01:15.490
DDL Creation Time	0:00:33.944
Total Run Time	0:01:49.444
Rows Inserted per sec	210,157
Data Generated (MB) per...	16.3

The creation of the schema appears to have been successful.

Status	Object Name
Valid	ORDERS, ORDER_ITEMS, CUSTOMERS, WAREHOUSES, ORDERENTRY_METADATA, INVENTORIES, PRODUCT_INFORMATION, PRODUCT_DESCRIPTIONS, ADDRESSES, CARD_DETAILS, PRD_DESC_PK, PROD_NAME_IX, PRODUCT_INFORMATION_PK, PROD_SUPPLIER_IX, PROD_CATEGORY_IX, INVENTORY_PK, INV_PRODUCT_IX, INV_WAREHOUSE_IX, ORDER_PK, ORD_SALES_REP_IX, ORD_CUSTOMER_IX, ORD_ORDER_DATE_IX, ORD_WAREHOUSE_IX, ORDER_ITEMS_PK, ITEM_ORDER_IX, ITEM_PRODUCT_IX, WAREHOUSES_PK,

Oracle Entry Install Wizard : 2.6.0.1137

File Log Viewer

Building the "Simple Order Entry" Schema while....

Inserting data into table ORDERS\_1

Rows being inserted per second = 261,1

16 of 16 Threads currently running

Data generation is 9.52% complete

# Tools: Swingbench

SwingBench 2.6.0.1137 (//10.22.21.187/plug1)

File Help

Configuration Preferences Output Events

User Details Connection Pooling Properties

Username: soe

Password: [masked]

Connect String: [masked]

Driver Type: Oracle jdbc Driver

Connect to Oracle Cloud Service

Credential File: [empty]

Collect Database Statistics

Take AWR Snapshot at Start and End

Admin Username: [empty]

Admin Password: [masked]

Load Environment Variables Distributed Controls

Number of Users: 100

Min. Inter Delay Between Transactions (ms): 0

Max. Inter Delay Between Transactions (ms): 0

Min. Intra Delay Within Transactions (ms): 0

Max. Intra Delay Within Transactions (ms): 0

.ogon Delay (milliseconds): 0

.ogon Group: 1

.wait Till All Sessions Log On: false

.ogoff Post Transaction: false

Tx. per Reconnect: 0

Benchmark Run Time (hh:min): 0 0

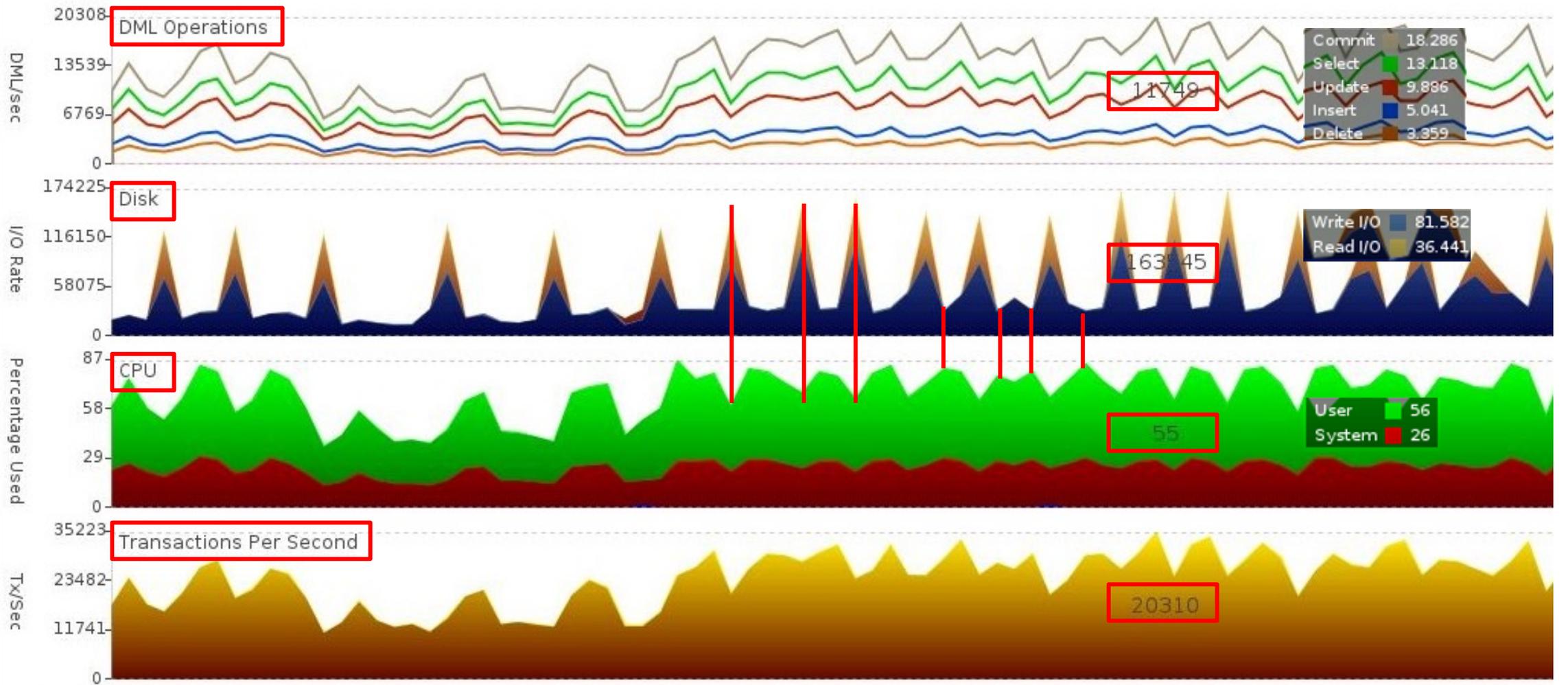
Transactions Jobs

Id	Class Name	Short Name	Load Ratio	Activate ?
Insert Transaction	com.dom.benchmarking.swingbench.stresstest.StressTestInsert	I	15	<input checked="" type="checkbox"/>
Simple Select	com.dom.benchmarking.swingbench.stresstest.StressTestSelect	S	40	<input checked="" type="checkbox"/>
Update Transaction	com.dom.benchmarking.swingbench.stresstest.StressTestUpdate	U	30	<input checked="" type="checkbox"/>
Delete Transaction	com.dom.benchmarking.swingbench.stresstest.StressTestDelete	D	10	<input checked="" type="checkbox"/>

Chart Preferences Logged on Sessions 100/100

Stress Test

Users Logged On : 100



Top ADDM Findings by Average Active Sessions

Finding Name	Avg act	Percen	Task Name
Commits and Rollbacks	80.14	83.57	ADDM:4269972545_1_35494
CPU Usage	80.14	5.46	ADDM:4269972545_1_35494
PL/SQL Execution	80.14	3.05	ADDM:4269972545_1_35494
Top SQL Statements	80.14	2.55	ADDM:4269972545_1_35494
Log File Switches	80.14	2.09	ADDM:4269972545_1_35494
Load Profile			
	Per Second	Per Transaction	Per Exec Per Call
DB Time (s):	80.1	0.1	0.00 0.01
DB CPU(s):	6.0	0.0	0.00 0.00
Background CPU(s):	0.2	0.0	0.00 0.00
Redo size (bytes):	5,640,290.4	3,788.9	
Logical read (blocks):	797,236.6	535.6	
Block changes:	36,039.1	24.2	
Physical read (blocks):	4.7	0.0	
Physical write (blocks):	4,014.2	2.7	
Read IO requests:	3.4	0.0	
Write IO requests:	2,926.9	2.0	
Read IO (MB):	0.0	0.0	
Write IO (MB):	31.4	0.0	
IM scan rows:	0.0	0.0	
Session Logical Read IM:	0.0	0.0	
User calls:	7,053.1	4.7	
Parases (SQL):	2,627.0	1.8	
Hard parases (SQL):	0.1	0.0	
SQL Work Area (MB):	40.6	0.0	
Logons:	0.1	0.0	
User logons:	0.0	0.0	
Executes (SQL):	37,296.1	25.1	
Rollbacks:	13.1	0.0	
Transactions:	1,488.6		

run swingbench



Top ADDM Findings by Average Active Sessions

Finding Name	Avg act	Percen	Task Name
Commits and Rollbacks	83.70	84.92	ADDM:4269972545_1_35498
Log File Switches	83.70	2.17	ADDM:4269972545_1_35498
PL/SQL Execution	83.70	1.85	ADDM:4269972545_1_35498
Load Profile			
	Per Second	Per Transaction	Per Exec Per Call
DB Time (s):	83.7	0.1	0.00 0.01
DB CPU(s):	5.1	0.0	0.00 0.00
Background CPU(s):	0.2	0.0	0.00 0.00
Redo size (bytes):	4,771,660.3	3,788.1	
Logical read (blocks):	637,768.0	506.3	
Block changes:	30,407.0	24.1	
Physical read (blocks):	2.9	0.0	
Physical write (blocks):	3,461.2	2.8	
Read IO requests:	2.6	0.0	
quests:	2,451.5	2.0	
O (MB):	0.0	0.0	
O (MB):	27.0	0.0	
n rows:	0.0	0.0	
ead IM:	0.0	0.0	
calls:	5,934.3	4.7	
(SQL):	2,222.1	1.8	
(SQL):	0.0	0.0	
a (MB):	34.5	0.0	
Logons:	0.0	0.0	
SQL logons:	0.0	0.0	
Executes (SQL):	31,453.6	25.0	
Rollbacks:	11.2	0.0	
Transactions:	1,259.6		

create snapshot

stop swingbench



Top ADDM Findings by Average Active Sessions

Finding Name	Avg act	Percen	Task Name
Commits and Rollbacks	80.53	81.64	ADDM:4269972545_1_35497
CPU Usage	80.53	5.35	ADDM:4269972545_1_35497
Top SQL Statements	80.53	5.13	ADDM:4269972545_1_35497
PL/SQL Execution	80.53	4.94	ADDM:4269972545_1_35497
Log File Switches	80.53	2.11	ADDM:4269972545_1_35497
Load Profile			
	Per Second	Per Transaction	Per Exec Per Call
DB Time (s):	80.6	0.1	0.00 0.01
DB CPU(s):	5.9	0.0	0.00 0.00
Background CPU(s):	0.2	0.0	0.00 0.00
Redo size (bytes):	6,092,463.5	3,884.4	
Logical read (blocks):	789,418.2	502.7	
Block changes:	39,083.7	24.3	
Physical read (blocks):	3.6	0.0	
Physical write (blocks):	4,334.1	2.8	
Read IO requests:	3.0	0.0	
Write IO requests:	3,330.1	2.1	
Read IO (MB):	0.0	0.0	
Write IO (MB):	33.9	0.0	
IM scan rows:	0.0	0.0	
Session Logical Read IM:	0.0	0.0	
User calls:	7,405.2	4.7	
Parases (SQL):	2,765.6	1.8	
Hard parases (SQL):	0.0	0.0	
SQL Work Area (MB):	41.6	0.0	
Logons:	0.1	0.0	
User logons:	0.0	0.0	
Executes (SQL):	39,235.1	25.0	
Rollbacks:	13.8	0.0	
Transactions:	1,568.5		

change Data Guard configuration

create snapshot

Top ADDM Findings by Average Active Sessions

Finding Name	Avg act	Percen	Task Name
Commits and Rollbacks	84.33	85.82	ADDM:4269972545_1_35499
Top SQL Statements	84.33	4.40	ADDM:4269972545_1_35499
PL/SQL Execution	84.33	1.77	ADDM:4269972545_1_35499
Load Profile			
	Per Second	Per Transaction	Per Exec Per Call
DB Time (s):	84.3	0.1	0.00 0.01
DB CPU(s):	5.0	0.0	0.00 0.00
Background CPU(s):	0.2	0.0	0.00 0.00
Redo size (bytes):	4,643,194.7	3,880.8	
Logical read (blocks):	619,066.6	512.4	
Block changes:	29,025.3	24.3	
Physical read (blocks):	2.9	0.0	
Physical write (blocks):	3,353.9	2.8	
Read IO requests:	2.5	0.0	
Write IO requests:	2,489.5	2.1	
Read IO (MB):	0.0	0.0	
Write IO (MB):	26.2	0.0	
IM scan rows:	0.0	0.0	
Session Logical Read IM:	0.0	0.0	
User calls:	5,632.6	4.7	
Parases (SQL):	2,111.4	1.8	
Hard parases (SQL):	0.0	0.0	
SQL Work Area (MB):	36.4	0.0	
Logons:	0.1	0.0	
User logons:	0.0	0.0	
Executes (SQL):	29,816.1	24.9	
Rollbacks:	10.5	0.0	
Transactions:	1,196.5		

- nmon (**N**igel's performance **M**onitor for Linux) – das bessere top
- Installation: `yum install nmon`
- zwei Betriebsmodi: interactive und capture
- capture erzeugt Textfile `<hostname>_YYYYMMDD_HHMM.nmon`
- im Textfile sind Messwerte kommasepariert gespeichert
- z.B. Loadwerte für die CPU (T0010 ist dann die 10. Messung):  
`CPU001, T0010, 1.9, 1.0, 43.8, 53.5, 0.0`
- mit Env-Variable NMON Vorbelegung beim Output möglich: `NMON="cdt5-"`

- NMON="cdt5-" 5: top processes order by IO - : halbes time for screen refresh

```
For help type H or ...
nmon -? - hint
nmon -h - full details

To stop nmon type q to Quit

-

Oracle Linux Server release 8.5 VERSION="8.5"
Vendor=AuthenticAMD Model=AMD EPYC 7402 24-Core Processor
MHz=2794.749 bogomips=5589.49 lscpu:CPU=8 Little Endian
ProcessorChips=1 PhysicalCores=8 Sockets=1 Cores=8 Thrds=1
VirtualCPUs =8 MHz=2794 max=0 min=0

Use these keys to toggle statistics on/off:
c = CPU l = CPU Long-term - = Faster screen updates
C = " WideView U = Utilisation + = Slower screen updates
m = Memory V = Virtual memory j = File Systems
d = Disks n = Network . = only busy disks/procs
r = Resource N = NFS h = more options
k = Kernel t = Top-processes q = Quit
```





- **Tools zur graphischen Aufbereitung von „nmon“ – Daten, u.a.:**
  - **NMONVisualizer**
  - **nmon Analyser Excel Spreadsheet**
  - **nmonchart**
- **nmonchart ist ein Shell-Skript, erzeugt ein html - Dokument**
- **tar – File entpacken und loslegen: nmonchart <your-data>.nmon**
- **Graphen/Statistiken mit Zoom-Möglichkeit in kleinere Zeiträume (CPU, Network, Disk Read, Disk Write, etc.)**

CPU Util. CPU Use RunQ Blocked pSwitch ForkExec Memory Swap  
Network Net Packet Disk Busy Unstacked Disk Read Unstacked Disk Write Unstacked Disk BSize Disk Xfers Unstacked JFS



### Standby – ohne Standby-Redologs

1.) Motivation

2.) Herausforderungen

3.) Tools

4.) Zusammenfassung

- es hat sehr viel Spaß gemacht, obwohl ich viel geflucht habe
- viel über Data Guard gelernt, insbesondere über die Architektur (@ORACLE – ein neues Diagramm bitte)
- wenn sich die Möglichkeit ergibt, dann sammle Daten für jedes Projekt
- traue nicht den ersten Messergebnissen, sondern führe mehrere Messungen durch
- zwei Vorträge zu gestalten, die inhaltlich zusammengehören ist eine Herausforderung
- Danke Martin für die gute Zusammenarbeit, das Setup, die Geduld, die Sprüche und viel Gelächter

**Also:** Fortsetzung folgt – diesmal RAC statt Data Guard auf der K&A in Nürnberg

- **Data Guard Concepts and Administration:**

<https://docs.oracle.com/en/database/oracle/oracle-database/19/sbydb/oracle-data-guard-concepts.html#GUID-F78703FB-BD74-4F20-9971-8B37ACC40A65>

[https://docs.oracle.com/en/database/oracle/oracle-database/19/refrn/V-DATAGUARD\\_PROCESS.html#GUID-F07EA03A-F3EF-4A2E-9250-A28AF3CEF5CF](https://docs.oracle.com/en/database/oracle/oracle-database/19/refrn/V-DATAGUARD_PROCESS.html#GUID-F07EA03A-F3EF-4A2E-9250-A28AF3CEF5CF)

- **ORADEBUG:**

<http://www.juliandyke.com/Diagnostics/Tools/ORADEBUG/ORADEBUG.php>

- **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.doag.org/de/home/news/aufgezeichnet-die-oracle-datenbank-mittels-swingbench-besser-verstehen>

- **nmon und nmonchart:**

<http://nmon.sourceforge.net/pmwiki.php>

<http://nmon.sourceforge.net/pmwiki.php?n=Site.Nmonchart>



**axel.vomstein@bss.gmbh**

**martin.klier@performing-db.com**