



ANALYSE UND VISUALISIERUNG VON STATSPACK UND AWR DATEN

DOAG Konferenz 2012, Nürnberg

Marcus Mönnig

Lichtblick AG, Hamburg

Vorstellung

Marcus Mönnig, Informatiker (B.Sc.), OCP 10/11

E-Mail: mm@marcusmoennig.de

Internet: <http://marcusmonnig.wordpress.com/>

Berufliche Stationen der letzten 16 Jahre

Selbständiger Softwareentwickler

Angestellter Softwareentwickler

Oracle Consultant / DBA für ein mittelständiges Softwarehaus
(Software für die pharmazeutische Industrie)

Heute: DBA bei der Lichtblick AG (Ökologischer
Energieversorger)

Vorstellung „Mumbai“

What is Mumbai?

Mumbai is a freeware Windows application targeted at Oracle DBAs and consultants, with a special focus on performance analysis

Features

- Instant access to important database views vital for DBAs and consultants
- Retrieved data in Mumbai can be sliced and diced in various ways (sorting, filtering, grouping, group arithmetic, etc.)
- Retrieve alert.log and trace files from the database server to Mumbai and process/analyze them there.
- Start and stop 10046 traces for any session or process and trace the statements from your SQL console session with just a few clicks
- Powerful 10046 trace file viewer if you need to dig into the details of a TKPROF or OraSRP report. OraSRP reports can be generated from within Mumbai.
- Heap dump analyzer that aggregates data from trace files
- ...

Agenda

- Historie und Grundprinzip
- Unterschiede zwischen Statspack und AWR
- Probleme und Eigenheiten von Statspack und AWR
- Analysemöglichkeiten von Oracle
- Erweiterte Analysemöglichkeiten in Mumbai

Historie

- UTLBSTAT.SQL/UTLESTAT.SQL (vor Oracle 8i)
Begin und End Skripte zum manuellen Erzeugen von einzelnen textuellen Reports zwischen zwei Zeitpunkten

- Statspack (ab Oracle 8.1.6 bis heute)

SP

Historisierte, automatisierte Sammlung, Speicherung und Auswertung von Performance Daten

- Automatic Workload Repository (AWR) (seit Oracle 10g)

AWR

Weiterentwicklung von Statspack mit gleichbleibendem Prinzip, aber weiteren und teilweise „besseren“ Datenquellen

Grundprinzip von Statspack und AWR (I)

- Snapshots
Kopieren von Performance Daten in historisierte Tabellen in regelmäßigen Intervallen
- Beispiel: Quelle **V\$SGASTAT**

SP

```
INSERT INTO STATSPACK$SGASTAT
(SNAP_ID, DBID, INSTANCE_NUMBER, POOL, NAME, BYTES)
SELECT
:B4, :B3, :B2, POOL, NAME, BYTES FROM ( SELECT POOL, NAME, BYTES, 100*(BYTES)/(SUM(BYTES) OVER (PARTITION BY
POOL)) PART_PCT
FROM V$SGASTAT)
WHERE PART_PCT >= :B1 OR POOL IS NULL OR NAME = 'free memory'
```

AWR

```
INSERT INTO WRH$_SGASTAT
(SNAP_ID, DBID, INSTANCE_NUMBER, POOL, NAME, BYTES)
SELECT
:SNAP_ID, :DBID, :INSTANCE_NUMBER, POOL, NAME, BYTES FROM (SELECT POOL, NAME, BYTES, 100*(BYTES) /
(SUM(BYTES) OVER (PARTITION BY POOL)) PART_PCT
FROM V$SGASTAT)
WHERE
PART_PCT >= 1 OR POOL IS NULL OR NAME = 'free memory'
ORDER BY NAME, POOL
```

Grundprinzip von Statspack und AWR (III)

- Analyse der gesammelten Daten
 - Differenzbildung zwischen zwei Snapshots
 - Verlauf über einen Zeitraum betrachten

Unterschiede zwischen Statspack und AWR (I)

- SP: kostenfrei, PL/SQL Quellcode, Edition-übergreifend
AWR: Teil des kostenpflichtigen Diagnostic Pack, nur in Oracle Enterprise Edition, in C implementiert
- AWR: Top-SQL aus Active Session History (ASH) Daten
STATSPACK: Aggregierte Daten aus V\$SQL
=> AWR liefert deutlich aussagekräftigere Daten
- AWR: Textuelle Vergleichsreports für zwei Intervalle
- AWR speichert aus den Quell-Performanceviews **viele** Spalten, die Statspack nicht speichert
- Statspack speichert aus den Quell-Performanceviews **einige** Spalten, die AWR nicht speichert (z.B. PROGRAM_ID und PROGRAM_LINE# aus V\$SQL)

Unterschiede zwischen Statspack und AWR (II)

- AWR: Import/Export Skripte für alle/ausgewählte Snapshot-Intervalle
- AWR zeichnet Metriken auf
 - WRH\$_FILEMETRIC_HISTORY
 - WRH\$_SESSMETRIC_HISTORY
 - WRH\$_SYSMETRIC_HISTORY
 - WRH\$_WAITCLASSMETRIC_HISTORY
 - WRH\$_SYSMETRIC_SUMMARY
- AWR zeichnet Service-Statistiken auf
 - WRH\$_SERVICE_NAME
 - WRH\$_SERVICE_STAT
 - WRH\$_SERVICE_WAIT_CLASS
- Statspack zeichnet Oracle Streams Performance Views auf
 - STATS\$PROPAGATION_SENDER
 - STATS\$PROPAGATION_RECEIVER

Unterschiede zwischen Statspack und AWR (III)

- Theorie: Im direkten Vergleich scheint AWR überlegen.
- Praxis: In den meisten Fällen lassen sich zurückliegende (Performance)Probleme mit Statspack-Daten genauso gut analysieren.
- Aber: AWR nutzt ASH Daten für die Top-SQL-Statement Analyse und liefert hier klar bessere Daten.

Agenda

- Historie und Grundprinzip
- Unterschiede zwischen Statspack und AWR
- Probleme und Eigenheiten von Statspack und AWR
- Analysemöglichkeiten von Oracle
- Erweiterte Analysemöglichkeiten in Mumbai

Probleme / Eigenheiten von Statspack und AWR (I)

- Analyse der gesammelten Daten muss in Betracht ziehen wie diese Daten gesammelt wurden
 - Qualität der Quelldaten und der gesammelten Daten
 - Intervall für die Datensammlung
 - Mehrfachzählung
 - Fehlende Bindevariablen
 - FILTER / ACCESS Prädikate in Ausführungsplänen bei Statspack

Probleme / Eigenheiten von Statspack und AWR (II)

Qualität der gesammelten Daten

- Analyseergebnis immer nur so gut wie die Daten die zur Verfügung stehen
- V\$SGASTAT Beispiel von oben:

SP

```
INSERT INTO STAT$SGASTAT
( SNAP_ID , DBID , INSTANCE_NUMBER , POOL , NAME , BYTES )
SELECT :B4 , :B3 , :B2 , POOL , NAME , BYTES FROM
( SELECT POOL , NAME , BYTES , 100*(BYTES)/(SUM(BYTES) OVER (PARTITION BY POOL)) PART_PCT FROM
V$SGASTAT)
WHERE PART_PCT >= :B1 OR POOL IS NULL OR NAME = 'free memory'
```

- Wie aussagekräftig sind die Quelldaten?
- Datenverlust durch Projektion? Selektion? Aggregation?
- Noch ein Beispiel: V\$SQL => STAT\$SQL_SUMMARY

Probleme / Eigenheiten von Statspack und AWR (II)

Qualität der gesammelten Daten

```
INSERT INTO STATSPACK_SUMMARY ( SNAP_ID , DBID , INSTANCE_NUMBER , ...)
```

```
SELECT
```

```
:B3 , :B2 , :B1 , MAX(SUBSTRB(SQL_TEXT,1,31)) TEXT_SUBSET , MAX(SQL_ID) SQL_ID , SUM(SHARABLE_MEM) SHARABLE_MEM ,  
SUM(SORTS) SORTS , MAX(MODULE) MODULE , SUM(LOADED_VERSIONS) LOADED_VERSIONS , SUM(FETCHES) FETCHES ,  
SUM(EXECUTIONS) EXECUTIONS , SUM(PX_SERVERS_EXECUTIONS) PX_SERVERS_EXECUTIONS , SUM(END_OF_FETCH_COUNT)  
END_OF_FETCH_COUNT , SUM(LOADS) LOADS , SUM(INVALIDATIONS) INVALIDATIONS , SUM(PARSE_CALLS) PARSE_CALLS ,  
SUM(DISK_READS) DISK_READS , SUM(DIRECT_WRITES) DIRECT_WRITES , SUM(BUFFER_GETS) BUFFER_GETS ,  
SUM(APPLICATION_WAIT_TIME) APPLICATION_WAIT_TIME , SUM(CONCURRENCY_WAIT_TIME) CONCURRENCY_WAIT_TIME ,  
SUM(CLUSTER_WAIT_TIME) CLUSTER_WAIT_TIME , SUM(USER_IO_WAIT_TIME) USER_IO_WAIT_TIME , SUM(PLSQL_EXEC_TIME)  
PLSQL_EXEC_TIME , SUM(JAVA_EXEC_TIME) JAVA_EXEC_TIME , SUM(ROWS_PROCESSED) ROWS_PROCESSED ,  
MAX(COMMAND_TYPE) COMMAND_TYPE , ADDRESS , MAX(HASH_VALUE) HASH_VALUE , OLD_HASH_VALUE , COUNT(1)  
VERSION_COUNT , SUM(CPU_TIME) CPU_TIME , SUM(ELAPSED_TIME) ELAPSED_TIME , NULL AVG_HARD_PARSE_TIME ,  
MAX(OUTLINE_SID) OUTLINE_SID , MAX(OUTLINE_CATEGORY) OUTLINE_CATEGORY , MAX(CHILD_LATCH) CHILD_LATCH ,  
MAX(SQL_PROFILE) SQL_PROFILE , MAX(PROGRAM_ID) PROGRAM_ID , MAX(PROGRAM_LINE#) PROGRAM_LINE# ,  
MAX(EXACT_MATCHING_SIGNATURE) EXACT_MATCHING_SIGNATURE , MAX(FORCE_MATCHING_SIGNATURE)  
FORCE_MATCHING_SIGNATURE , MAX(LAST_ACTIVE_TIME) LAST_ACTIVE_TIME
```

```
FROM V$SQL
```

```
WHERE
```

```
IS_OBSOLETE = 'N'
```

```
AND SQL_ID IN (SELECT SQL_ID FROM STATSPACK_STATS_SUMMARY SQLSTATS WHERE ( BUFFER_GETS > :B9 OR DISK_READS >  
:B8 OR PARSE_CALLS > :B7 OR EXECUTIONS > :B6 OR SHARABLE_MEM > :B5 OR VERSION_COUNT > :B4 ) )
```

```
GROUP BY OLD_HASH_VALUE, ADDRESS
```

Probleme / Eigenheiten von Statspack und AWR (III)

Qualität der gesammelten Daten – SUM(ELAPSED_TIME)

- V\$SQL enthält Daten über Child-Cursor
- STATSPACK_SUMMARY gruppiert über Parent-Cursor

V\$SQL bei Snapshot 1:

SQL_ID	OLD_HASH_VALUE	ADDRESS	CHILD_NUMI ▲	ELAPSED_TIME
adu5n2ua8qxt9	3.070.929.891	00000005F7046618	0	658.954
adu5n2ua8qxt9	3.070.929.891	00000005F7046618	1	186.473
adu5n2ua8qxt9	3.070.929.891	00000005F7046618	2	468.987
adu5n2ua8qxt9	3.070.929.891	00000005F7046618	3	187.830
adu5n2ua8qxt9	3.070.929.891	00000005F7046618	4	79.249
				1581493,00

V\$SQL bei Snapshot 2:

SQL_ID	OLD_HASH_VALUE	ADDRESS	CHILD_NU ▲ <input checked="" type="checkbox"/>	ELAPSED_TIME
adu5n2ua8qxt9	3.070.929.891	00000005F7046618	0	658.954
adu5n2ua8qxt9	3.070.929.891	00000005F7046618	1	186.473
adu5n2ua8qxt9	3.070.929.891	00000005F7046618	3	187.830
adu5n2ua8qxt9	3.070.929.891	00000005F7046618	4	79.249
				1112506,00

➡ Delta für ELAPSED_TIME zwischen S2 und S1: 1,11sec – 1,58sec = - 0,47sec

➡ In Reports tauchen negative, zu kleine oder zu große Werte auf.

Probleme / Eigenheiten von Statspack und AWR (IV)

Qualität der gesammelten Daten – MAX(MODULE)

```
SQL ordered by CPU DB/Inst: PSB/PSB Snaps: 8696-8697
-> Total DB CPU (s):          8,539
-> Captured SQL accounts for 232.5% of Total DB CPU
-> SQL reported below exceeded 1.0% of Total DB CPU
```

CPU Time (s)	Executions	CPU per Exec (s)	%Total	Elapsed Time (s)	Buffer Gets	Old Hash Value
4407.08	20,294	0.22	51.6	10006.66	1,228,369,784	3703299877

Module: JDBC Thin Client <= MAX(MODULE) über Parent Cursor

```
SELECT SEQ_THM FROM (SELECT CAST(SUBSTR(COLUMN_VALUE,1,20) AS CHAR(20)) THM_ID, CAST(SUBSTR(COLUMN_VALUE,21,4) AS CHAR(4)) SEQ_THM FROM TABLE(MAYO_RPS.GET_CUSTOMER_ORDER_SEQUENCES(:B1 , :B2 , :B3 , :B4 ))) WHERE THM_ID = :B5
```

3943.14	157,316	0.03	46.2	6915.60	1,034,202,723	1127338565
---------	---------	------	------	---------	---------------	------------

Module: sel_ancomm_vss_06.tsk@c2aixprod (TNS v1-v3) <= MAX(MODULE) über Parent Cursor

```
SELECT * FROM ( SELECT L.AREA, L.AISLE, L.X, L.Y, L.Z FROM LOCATION L, MHV_STOCK_LEVEL M, LOC_GROUP LG WHERE L.AREA = 'AKL_' AND L.AISLE = :B1 AND L.X LIKE '00__' AND L.Y LIKE '00__' AND L.Z LIKE '00_' AND L.LOC_GROUP_ID LIKE 'AISLE%' AND L.CNT_THM_IN >=
```


Probleme / Eigenheiten von Statspack und AWR (V)

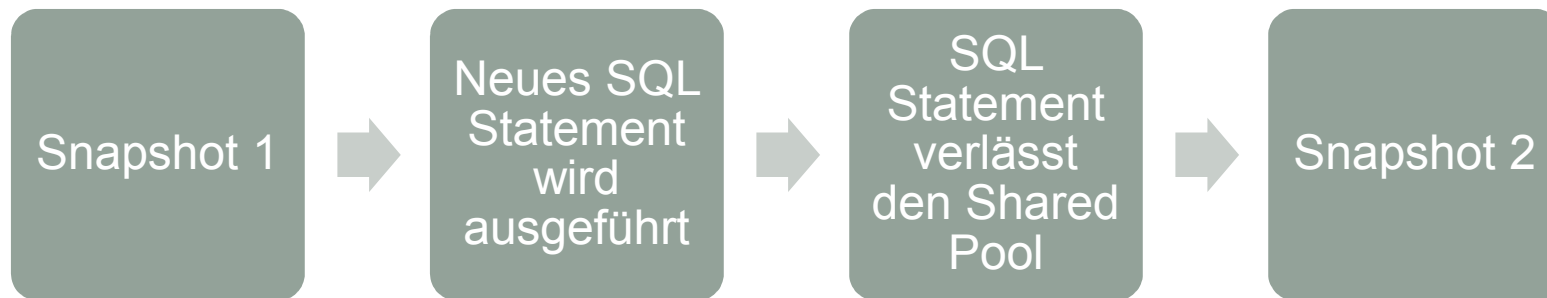
Qualität der gesammelten Daten

- SP: 11 von 55 INSERT AS SELECT Statements enthalten GROUP BY Aggregationen
 - AWR: 15 von 74 INSERT AS SELECT Statements enthalten GROUP BY Aggregationen
 - Vielzahl von Selektions-/Projektionseinschränkungen
- ➔ Über PL/SQL Quellcode (SP) oder 10046 extended SQL trace Statements analysieren

Probleme / Eigenheiten von Statspack und AWR (VI)

Größe des Snapintervalls

- Kurze Intervalle => bessere zeitliche Einordnung von DB-Aktivität oder Problemen möglich
- Lange Intervalle ungeeignet für SQL-Statement-Analyse

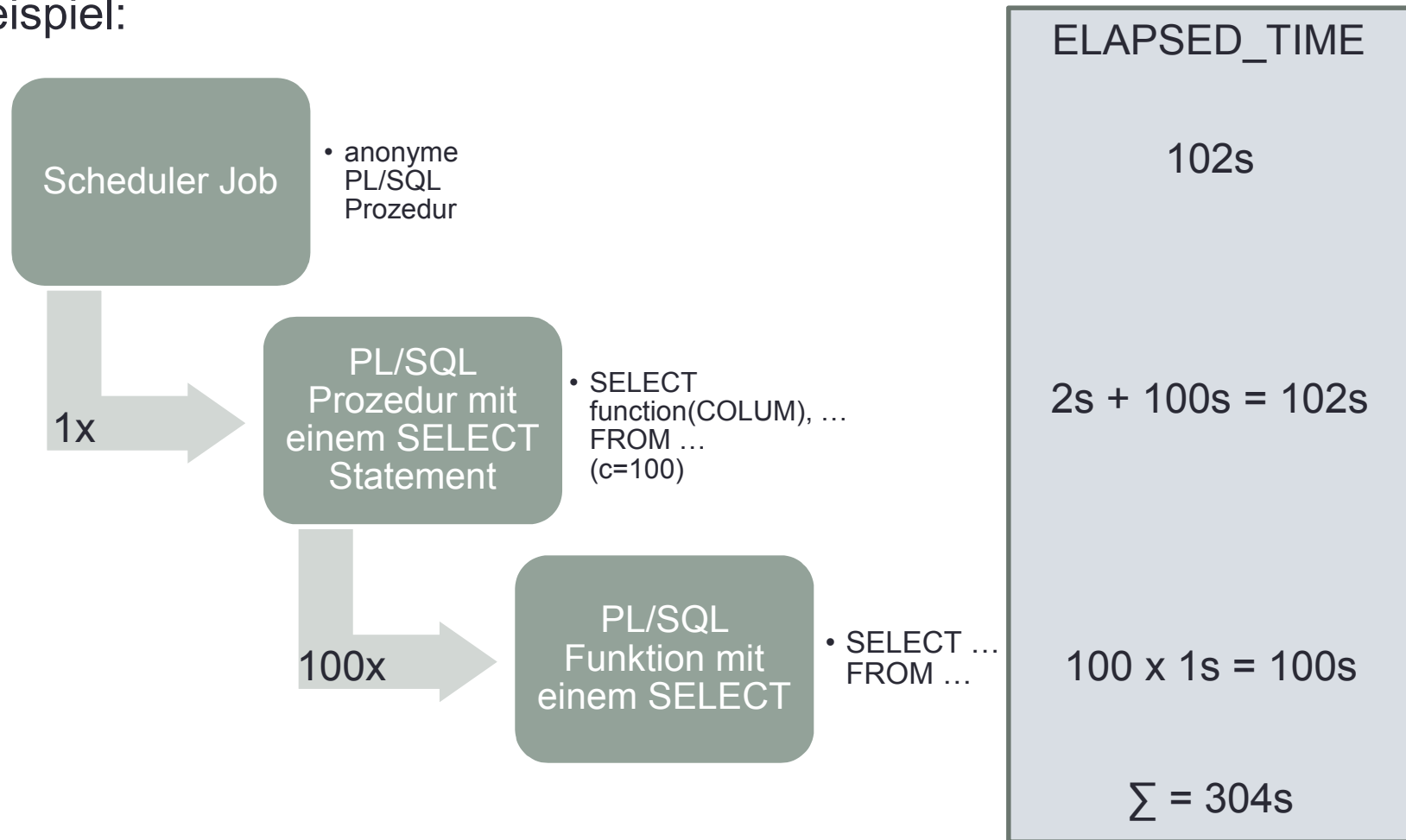


➔ SQL Statement taucht in keinem Report auf

Probleme / Eigenheiten von Statspack und AWR (VII)

Mehrfachzählung

- Zeiten von SQL Statements werden u.U. mehrfach erfasst
- Beispiel:



Probleme / Eigenheiten von Statspack und AWR (IIX)

Fehlende Bindevariablen

- Hohe Anzahl semantisch gleicher SQL Statements ohne Bindevariablen

- | | |
|---|---|
| • SELECT ... FROM ... WHERE ID=1 | SQL_ID = 6kqmn57kd2yd8 |
| • SELECT ... FROM ... WHERE ID=2 | SQL_ID = b3qyup0nma9fs |
| • SELECT ... FROM ... WHERE ID=3 | SQL_ID = ahqsp2cg1kyky |
| • SELECT ... FROM ... WHERE ID=... | SQL_ID = ... |
| • Statt: SELECT ... FROM ... WHERE ID=:ID | SQL_ID unabhängig von Wert der Bindevariablen |

 Statements tauchen gar nicht als TOP-Statements auf, bzw. sind in ihrer Gesamtheit nicht als kritisch erkennbar

- Lösung: Vor der Auswertung nach FORCE_MATCHING_SIGNATURE gruppieren/aggregieren
- Aber: Sind Einzelstatements alle in STATSQL_SUMMARY erfasst?

Probleme / Eigenheiten von Statspack und AWR (IX)

FILTER/ACCESS Prädikate in Ausführungsplänen

- In von Statspack aufgezeichneten Ausführungsplänen (Snap_level>5) fehlen FILTER/ACCESS Prädikate
- Package STATSPACK:

```
...  
      , 0 -- should be max(sp.access_predicates) (2254299)  
      , 0 -- should be max(sp.filter_predicates)  
...
```

Probleme / Eigenheiten von Statspack und AWR (X)

Empfehlungen

- Trotz der Eigenheiten: AWR und Statspack sind sehr nützliche Tools
- Hinterfragen Sie die Statspack/AWR Daten und die Reports!
 - Analyse des PL/SQL Quellcodes, Tracen der Snapshots-Operationen
 - Analyse des Applikationsquellcode bzw. Zusammenarbeit mit den Entwicklern
- Überprüfen Sie die Daten mit anderen Mitteln:
 - Live-Analyse von V\$... Views
 - Active Session History oder Snapper-Skript von Tanel Poder
 - Eigene Snapshots relevanter V\$... Views („Snap_Anything“ Package)

Agenda

- Historie und Grundprinzip
- Unterschiede zwischen Statspack und AWR
- Probleme und Eigenheiten von Statspack und AWR
- **Analysemöglichkeiten von Oracle**
- **Erweiterte Analysemöglichkeiten in Mumbai**

Analysemöglichkeiten von Oracle

	AWR Skript	Statspack Skript
Standard Datenbank-Report	awrrpt.sql	spreport.sql
Standard SQL-Report	awrsqrpt.sql	sprepsql.sql
Vergleichender Datenbank-Report	awrddrpt.sql	-
I/O Intensitäts-Report	spawrio.sql	-

Statspack

- Reportgenerierung nur über SQL*Plus

AWR

- Reportgenerierung in PL/SQL bzw. C implementiert, also auch ohne SQL*Plus nutzbar
- Teilweise Darstellung über den Enterprise Manager
- Weitere ASH Reports über historischen Daten

Erweiterte Analysemöglichkeiten in Mumbai

- Textuellen Reports basieren auf Deltas bzw. Absolutwertvergleichen zwischen Anfangs- und End-Snapshot – dazwischenliegende Snapshots werden ignoriert
- Detailinformationen im zeitlichen Verlauf gewünscht => Große Anzahl von textuellen Reports
- Textuelle Reports: Daten aus unterschiedlichen Quellen ohne zeitlichen Verlauf
- „Reports“ in Mumbai: Daten aus jeweils einer Quelle im zeitlichen Verlauf



Erweiterte Analysemöglichkeiten in Mumbai

- Demo

Erweiterte Analysemöglichkeiten in Mumbai

Korrelationen

- Daten bereits geladener Reports werden in zeitliche Korrelation gesetzt
- Korrelation anschaulich: Ähnlichkeit zweier Graphen von statistischen Reihen
- Ausgedrückt in prozentualer Korrelationzahl

- Rein zeitliche Korrelation, möglicherweise ohne logischen oder Ursache-Wirkung-Zusammenhang
- Geeignet für:
 - Aufrufhierarchien lassen sich schneller erfassen
 - Zuordnung eher ungewöhnlicher wait events zu SQL Statements
 - ...

Danke für Ihr Interesse und viel Spaß auf der DOAG 2012!

- **Quellen** (Google Suchworte **fett** markiert)
 - **Marcus** Mönning's Oracle and **Mumbai** Blog - <http://marcusmonnig.wordpress.com>
 - **Tanel** Poder's Session **Snapper** - <http://tech.e2sn.com/oracle-scripts-and-tools/session-snapper>
 - Marcus Mönning's **Snap Anything** Package - http://marcusmonnig.wordpress.com/2012/09/10/snap_anything-tiny-little-solution-for-snapping-and-recording-anything-that-you-can-query/

- **Kontakt**daten

Marcus Mönning

E-Mail:

mm@marcusmoennig.de

Internet:

<http://marcusmonnig.wordpress.com/>