Competence gainfully utilized

KOMPETENZ GEWINNBRINGEND EINSETZEN.

DOAG Conference 2016

Robert Barić

16.11.2016 17:00, Nürnberg Convention, Shanghai

RAT-less Replay

Who AM I

- Robert Barić
  - Oracle KnowHow since 9i/10g
  - Development Know How
  - 6 years with ITGAIN
  - Specialized in Performance Tuning and Backup/Recovery
  - DB2 KnowHow 9.1 – 10.1

Who are we ?

- ITGAIN www.itgain.de
  - **IT** {Informationstechnologie} & **GAIN** {Vorteil; Nutzen; Gewinn}
- Development & Integration
- BI und Data Warehouse
- SOA Competence Center
- Middleware
- Datenmanagement

About the Lecture

- You may ask questions during the lecture
- Tell if you can‘t hear or understand me
- You‘ll find additional information in the manuscript
- Only small parts of this topic can be covered here

AGENDA

- Why Replay?
- Use Cases
- RAT Scope of Functions
- Why RAT-less replay?
- Recording Options
- Trace Files – Features and Limts
- (Unified) Audit – Features and Limits
- Examples SQLReplayer

Why Record and Replay

Realworld Case: Movement to the cloud

- Real World example from a popular german car sharing company

- Synthetic Benchmarks tests passed all requirements
- Production system was implemented
- Result: Real Workload led to a desaster

- Conclusion: synthetical benchmarks are limited in their application

General usecase 1: Functional Tests

- Does your code still work on a different system?
- To Mind: How much function code do you cover?

- Changes with Hardware
  - I/O and Storage
  - CPU
  - Memory, Network
- Changes with Software like
  - OS version
  - DB version
  - Device drivers
  - patches

General usecase 2: Migration Costs and Time

- How much of your code still works on a different database ?
  - How many lines of code you need to change ?
  - What kind of code you need to change?

- Identifiy obstacles like
  - Different SQL syntax
  - Different semantic
  - Different behaviour (commit / rollback)
  - Different types (timestamp, date,..)

General Usecase: Performance Tests

- How does your workload perform

- Overall performance time, IO, CPU
- Performance with changed Workload
  - Multiply Workload during replay (more session, more SQLs)
  - Remove Wait Time

RAT Scope of Functions

RAT Supported		NOT RAT Supported	
Complete SQL Code (DML,DDL, PL/SQL)	✓	Direct Path Loads, Import/Export	✗
Local Transactions	✓	Flashback (DB, Query)	✗
Sessionchange	✓	Distributed Connections	✗
Connects/Disconnects	✓	Shared Server	✗
Limited PL/SQL RPCs	✓	Streams (non PL/SQL based AQ)	✗
Complete Lob Functionality	✓	OCI Object Navigation / REF-Binds	✗

Why RAT-Less Replay

- EE Option with Extra Cost
- No Support for Standard Edition
- Limited to Oracle Database

- Powerfull and feature rich, but
  - Restore to SCN needed
  - Record Files are „closed source“
  - For simple or smal workloads rather complicated

How to record a workload

Recording Options with Oracle (Q)

- Ordinary Views / Diagnostic Views
  - AWR, v$session, v$sql_text, V_$SQL_BIND_CAPTURE, etc..
- Ordinary Auditing
  - AUDIT SELECT TABLE, UPDATE TABLE, INSERT TABLE, DELETE TABLE
- FGA (Fine Grained Auditing)
- Unified Auditing
- Trace (10046 TRACE, SQL_TRACE, DBMS_MONITOR, ..)
- DB Triggers and manual Logging
- JDBC Interface Logging
- Logminer
  - The LogMiner Utility (Doc ID 62508.1)
  - Oracle 10g New Features Of LogMiner (Doc ID 249001.1)
- Streams
- ….

Challenges and Obstacles

Recording Options	Challenges/Obstacles
Diagnose / Tuning Options	Limited Sampling Rate, Limited Recording
Unified Auditing	Starting from 12c, Variable 4000 Byte limit, asynchron IO possible, conditional auditing
Standard Auditing	Extend of auditing may be limited, No Commit/Rollback 4000 Byte Limit with variables, No Lob/Clob variable support
FGA (Fine Grained Auditing)	EE Option, mostly same limits as ordinary auditing
Trace (10046 TRACE),	One file per session, possibly high overhead, 4000 byte limit
JDBC Interface Logging	Client side access, Overhead using multiple clients
Logminer	„Transformed“ SQLs and DDL. Some actions not available.
(Table) Trigger	Extra effort to implement. Because of needed DB changes possibly not useable

EVALUATION

Unified Auditing promises a balance between performance and available filters

- Available from version 12c
- No metrics except execution time
- No SQL depth information
- DML, DDL, PL/SQL, commit, rollback, …
- Bindvariablen restriction 4000 bytes

Trace Files

- Available for all DB versions
- Overhead can be high
- One file for each session
- More limits on binds variables (No Lobs)

EVALUATION

Asset	Unified Audit	Trace Files	Std. Auditing
Asynchron Writes	+	- (?)	-
Monitor commit/rollback	+	+	-
Mointor Failed SQL	+	+ (?)	+
Monitor Transactions	+	+	-
SQL Depth	-	+	-
Performance Messurement	-	+	-
LOB Binds	+	-	(+?)
Filter Actions	**Conditional audit**, Actions, User	Session, trigger, conditional on Appl. Information	User, Action

RAT-Less Scope of functions

RAT Supported		NOT RAT Supported	
Complete SQL Code (DML,DDL, PL/SQL)	✓	Direct Path Loads, Import/Export	✓ (UA)
Local Transactions	✓	Flashback (DB, Query)	✗
Session change	✓	Distributed Connections	(?)
Connects/Disconnects	✓	Shared Server	✓ (UA)
Limited PL/SQL RPCs	✓ (?)	Streams (non PL/SQL based AQ)	✗
Complete Lob Functionality	✗	OCI Object Navigation / REF-Binds	?

TRACE FILES: WHAT WE GET

- TIME: SCN, Event_Timestamp (tim), Elapsed Time (e in µs)
- CONTENT: SQL Text, SQL_BINDS, Hash Value (hv),
- MESSUREMENT: CPU Waits (c in µs), IO (p phy, blocks read), Network

- META INFORMATION DEP (Appl. Depth), rows returned (r)

- Limits: Limited trace filter (Session, Sql_Id)
- Limits: No Lobs, Refs, User Definied Types

ENABLE TRACE

```
DBMS_MONITOR.SESSION_TRACE_ENABLE(
        session_id    IN BINARY_INTEGER DEFAULT NULL,
        serial_num    IN BINARY_INTEGER DEFAULT NULL,
        waits         IN BOOLEAN DEFAULT TRUE,
        binds         IN BOOLEAN DEFAULT FALSE,
        plan_stat     IN VARCHAR2 DEFAULT NULL);
```

```
execute dbms_support.start_trace_in_session(:sid,:serial,waits=>true,binds=>false);
```

```
execute dbms_monitor.serv_mod_act_trace_enable('Service1', 'Module1', 'Action1',
waits => (TRUE|FALSE), binds => (TRUE|FALSE), instance_name => 'ORCL' );
```

```
alter session set events '10046 trace name context forever, level 4'
alter session set sql_trace=true;
```

TRACE FILE SAMPLE

```
CLOSE #140508518321392:c=0,e=1,dep=1,type=3,tim=1665434199619
=====================
PARSING IN CURSOR #140508518128040 len=32 dep=0 uid=215 oct=2 lid=8 tim=1665434199661 hv=3210460697 ad='b4a598c0' sqlid='avh7nuqzprght'
 insert into t2 values (:n1,:n2)
END OF STMT
PARSE #140508518128040:c=4000,e=1760,p=0,cr=12,cu=0,mis=0,r=0,dep=0,og=1,plh=0,tim=1665434199660
BINDS #140508518128040:
 Bind#0
  oacdty=01 mxl=128(60) mxlc=00 mal=00 scl=00 pre=00
  oacflg=03 fl2=1000000 frm=01 csi=873 siz=160 off=0
  kxsbbbfp=7fcab0515318  bln=128  avl=12  flg=05
  value="Robert Baric"
 Bind#1
  oacdty=96 mxl=32(24) mxlc=00 mal=00 scl=00 pre=00
  oacflg=03 fl2=1000000 frm=01 csi=873 siz=0 off=128
  kxsbbbfp=7fcab0515398  bln=32  avl=08  flg=01
  value="Rob     "
EXEC #140508518128040:c=0,e=927,p=0,cr=1,cu=4,mis=0,r=1,dep=0,og=1,plh=0,tim=1665434200622
WAIT #140508518128040: nam='SQL*Net message to client' ela= 4 driver id=1650815232 #bytes=1 p3=0 obj#=-1 tim=1665434200680
WAIT #140508518128040: nam='SQL*Net message from client' ela= 237 driver id=1650815232 #bytes=1 p3=0 obj#=-1 tim=1665434200936
CLOSE #140508518128040:c=0,e=6,dep=0,type=1,tim=1665434200973
```

AUDITING – WHAT WE GET

- TIME: SCN, Event_Timestamp
- ENTITY: Session_Id, Entry_ID, Transactions_Id Statement_Id, Instance_ID
- CONTENT SQL, BIND Variables, Schema

- META-INFORMATION

  Username, USERHOST

  ACTION_NAME, AUDIT_TYPE

  AUDIT_OPTION, UNIFIED_AUDIT_POLICIES, FGA_POLICY_NAME, ADDITIONAL_INFO

- LIMITS: No Wait Times Measurement
  No distinction own and other code like triggers

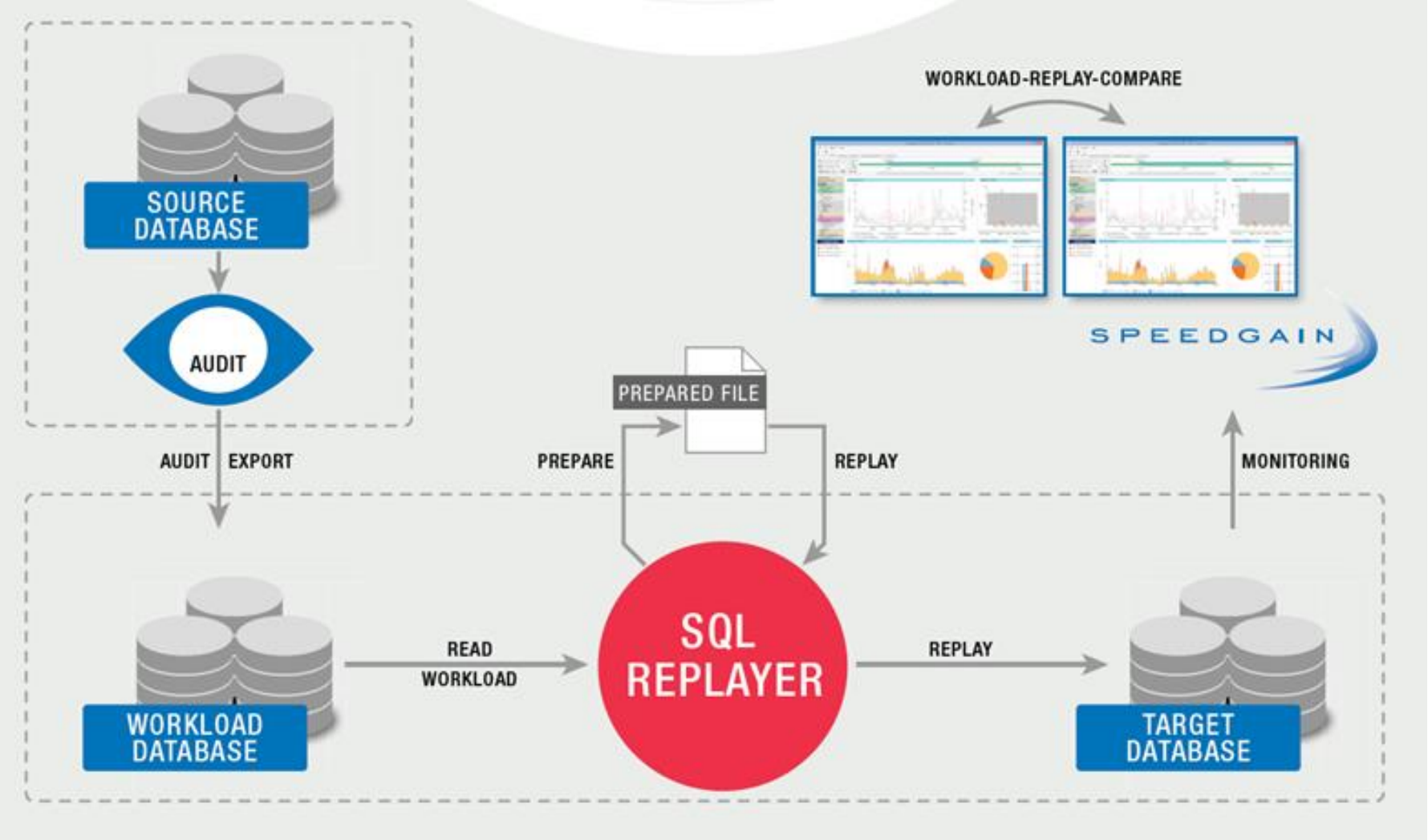
AUDIT FILE SAMPLE

STATEMENT_ID	EVENT_TIMESTAMP	ACTION_NAME	SCN	EXECUTION_ID	OBJECT_SCHEMA	OBJECT_NAME	SQL_TEXT	SQL_BINDS
12	09.11.16 15:42:40,238771000	CREATE TABLE	120682550	(null)	REPLAY1	T1B	create table t1b (int Integer,num number(9,2))	(null)
14	09.11.16 15:42:40,278608000	INSERT	120682573	(null)	REPLAY1	T1B	INSERT INTO T1B VALUES (:B2 , :B1)	#1(1):1 #2(1):2
16	09.11.16 15:42:40,278817000	INSERT	120682573	(null)	REPLAY1	T1B	INSERT INTO T1B VALUES (:B2 , :B1)	#1(2):23 #2(2):42
17	09.11.16 15:42:40,279069000	COMMIT	120682575	(null)	(null)	(null)	COMMIT	(null)
19	09.11.16 15:42:40,279329000	INSERT	120682575	(null)	REPLAY1	T1B	INSERT INTO T1B VALUES (:B2 , :B1)	#1(3):100 #2(3):200
21	09.11.16 15:42:40,279470000	INSERT	120682575	(null)	REPLAY1	T1B	INSERT INTO T1B VALUES (:B2 , :B1)	#1(4):2300 #2(4):4200
22	09.11.16 15:42:40,280106000	ROLLBACK	120682580	(null)	(null)	(null)	ROLLBACK	(null)

Recording Obstacles (Q)

- Datatypes are limited
- Additional Code may arise
  - Clear by rules
  - Clear by depth
- Impact on systems with high workload
  Business Impact
  synchron writes or limited buffer size
- Bugs
- …

REPLAY EXPLAINED VIA SQLREPLAYER



REPLAYING WITH SQLREPLAYER

- Config
  - IP, port, DB-name, username, (password), schema for workload and target database
  - Timestamp for begin and end of workload that you want to prepare
  - Define type of SQL Statements you want to prepare
    - ReadDML, WriteDML, DLL, CALL, OTHER, NONE
    - Include failures?
  - Directories for log and replay files (CSV, JSON)
  - Wait times for new connection and between SQL statements
  - Autocommit on or off
  - Eyecatching settings to easily identify replayed statements from others in monitoring solution
  - Loglevel (set to off for max performance)

REPLAYER PREPARE

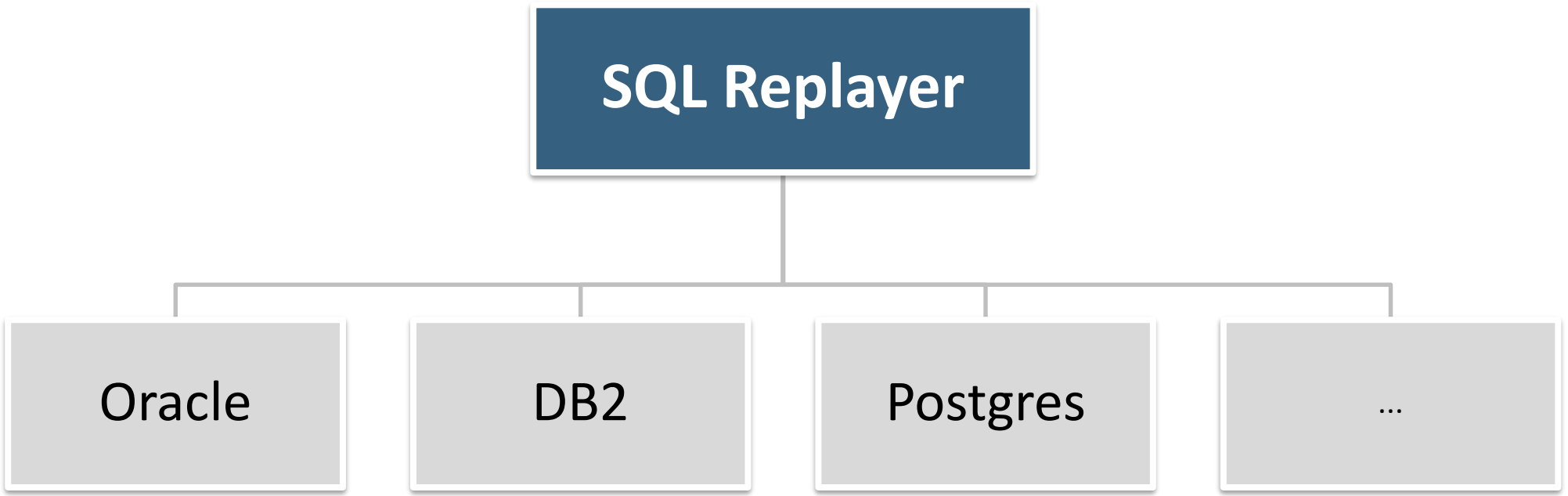
```
D:\SQLReplayer_V1.1.4_oracle_welldone>java -Xms256M -Xmx1024M -jar SQLReplayer.jar prepare
License holder: ITGAIN Consulting GmbH, Essener Str. 1, 30179 Hannover

Starting SQLReplayer Version 1.1.4
Recording
Record directory                                    [./replay_ora]
Logging in                                                [./log]
Run ID                                            [1478783795108]
Source DB                    [REPLAYDB@192.168.123.73:1521/DB12TST1]
Statements src                              [REPLAYDB.oracleexecute]
Read DML                                                   [true]
Write DML                                                 [false]
DDL                                                       [false]
CALL                                                      [false]
None                                                       [true]
Other                                                      [true]
Log Directory                                                [OK]
Init                                                         [OK]
DB Connection                                                [OK]
Statements to save                                         [444]


Recording      [.............................................]
```

SQL-REPLAYER REPLAY SUMMARY

REPLAY TO OTHER DATABASE SYSTEMS



Q & A

YOUR BUSINESS IS OUR CHALLENGE

ITGAIN GmbH
Essener Str.1
D-30173 Hannover

Telefon: +49 511 51 51 3700
Fax: +49 511 51 51 3800

Hamburg +49 40 284 175 50 Köln +49 221 650 78 299 Frankfurt +49 69 710 456 477 München +49 89 20802 7229