



# Flashback Database

Johannes Ahrends  
CarajanDB GmbH



**DOAG**  
Deutsche ORACLE-Anwendergruppe e.V.

- Oracle Spezialist seit 1992
  - 1992: Presales bei Oracle in Düsseldorf
  - 1999: Projektleiter bei Herrmann & Lenz Services GmbH
  - 2005: Technischer Direktor ADM Presales bei Quest Software GmbH
  - 2011: Geschäftsführer CarajanDB GmbH
- 2011 → Ernennung zum Oracle ACE
- Autor der Bücher:
  - Oracle9i für den DBA, Oracle10g für den DBA, Oracle 11g Release 2 für den DBA
- DOAG Themenverantwortlicher Datenbankadministration, Standard Edition
- Hobbies:
  - Drachen steigen lassen (Kiting) draußen wie drinnen (Indoorkiting)
  - Motorradfahren (nur draußen)





# Architektur

- Zurücksetzen der Datenbanken auf einen älteren Zustand
  - Bis Oracle 11g: gesamte Datenbank
  - Oracle 12.1: CDB und alle PDBs
  - Oracle 12.2: einzelne PDBs (Voraussetzung: Local Undo)
  - Ausnahmen bestätigen die Regel!
- Hat technisch nichts mit Flashback Table, Flashback Query, Flashback Data Archive, etc. zu tun!
- Voraussetzung:
  - ARCHIVE LOG MODE
  - FAST\_RECOVERY\_AREA
  - Enterprise Edition

# GUARANTEED RESTORE POINT

- Festlegen eines exakten Zeitpunkt, auf den (und nur auf den) die Datenbank zurückgesetzt werden kann
- Flashback Logs werden so lange aufbewahrt, wie der Restore Point existiert
- Gefahr, dass die Fast Recovery Area vollläuft und damit die Datenbank hängt
- Kann jederzeit eingeschaltet bzw. gelöscht werden

```
SQL> CREATE RESTORE POINT <name> GUARANTEE FLASHBACK DATABASE;
```

```
SQL> CREATE RESTORE POINT <name>  
FOR PLUGGABLE DATABASE <pdbname> GUARANTEE FLASHBACK DATABASE;
```

# FLASHBACK DATABASE MODE

- Datenbank kann auf jeden beliebigen Zeitpunkt zurückgesetzt werden
- Zusätzliche Restore Points (auch Guaranteed) können gesetzt werden
- Benötigt eine Zeitangabe
  - `db_flashback_retention_target` (in Minuten, Default 1440 = 1 Tag)
- Instanzen müssen nicht gestoppt werden

```
SQL> ALTER DATABASE FLASHBACK ON;  
SQL> SELECT flashback_on FROM v$database;
```

```
FLASHBACK_ON
```

```
-----
```

```
YES
```

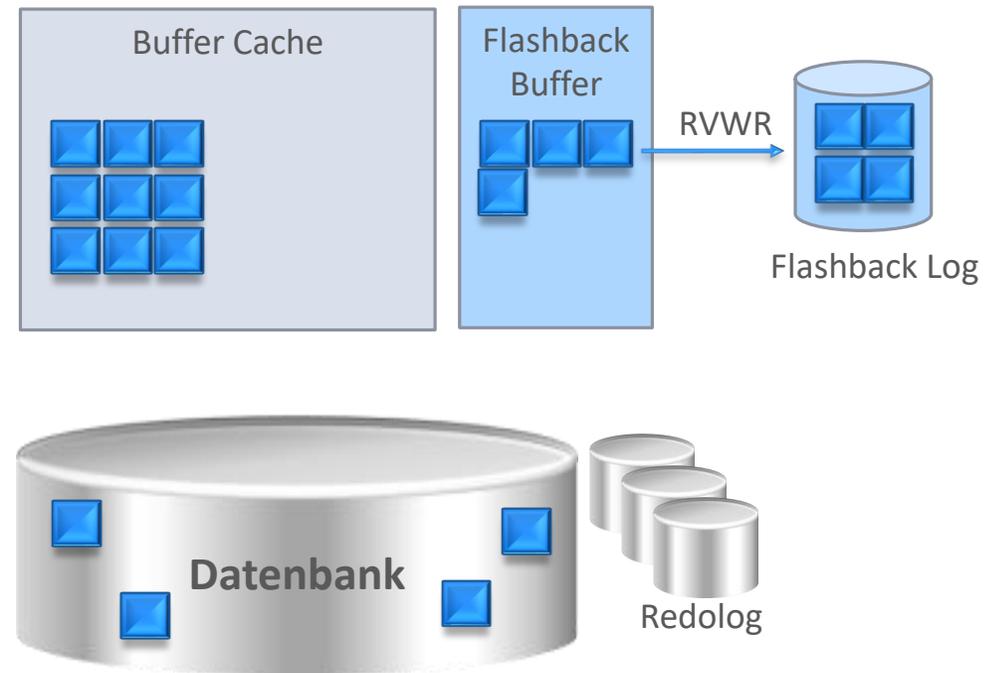
- FLASHBACK DATABASE Mode geht mit RMAN Restore oder Duplicate verloren
- Tablespace können vom Flashback Mode ausgenommen werden

```
SQL> ALTER TABLESPACE <name> FLASHBACK OFF;
```

- Warum das?
  - Tablespace mit großer Änderungsrate
  - Der Tablespace muss nach einem Flashback Database offline gesetzt werden

# Flashback Architektur (1)

1. Lesen eines Blockes aus der Datenbank
2. Ändern des Blockes  
→ geänderter Block wird in Flashback Buffer (FBL) kopiert
3. Ändern des gleichen Blockes  
→ keine Kopie!  
→ Block ist schon im FBL
4. Flashback Buffer wird mit dem RVWR (Recovery Writer) in ein Flashback Log geschrieben



- Flashback Buffer

```
SQL> SELECT * FROM v$sgastat  
      WHERE name LIKE 'flashback %';
```

POOL	NAME	BYTES	CON_ID
shared pool	flashback generation buff	4194304	1

- Änderungen an Blocks werden periodisch in die Flashback Buffer geschrieben
- RVWR Prozess schreibt Flashback Buffer in Flashback Logs (FRA)

```
SQL> SELECT name, log#, sequence#, bytes, first_change#, first_time, type  
FROM v$flashback_database_logfile;
```

NAME	LOG#	SEQUENCE#	BYTES	FIRST_CHANGE#	FIRST_TIME	TYPE
/u03/orabackup/LEONARD/flashback/o1_mf_dz64hhht_.flb	1	1	209715200	639310	27.10.2017 13:01:04	NORMAL
/u03/orabackup/LEONARD/flashback/o1_mf_dz64hkyh_.flb	2	2	209715200	811887	27.10.2017 13:11:28	NORMAL
/u03/orabackup/LEONARD/flashback/o1_mf_dz65330j_.flb	3	1	209715200	0		RESERVED

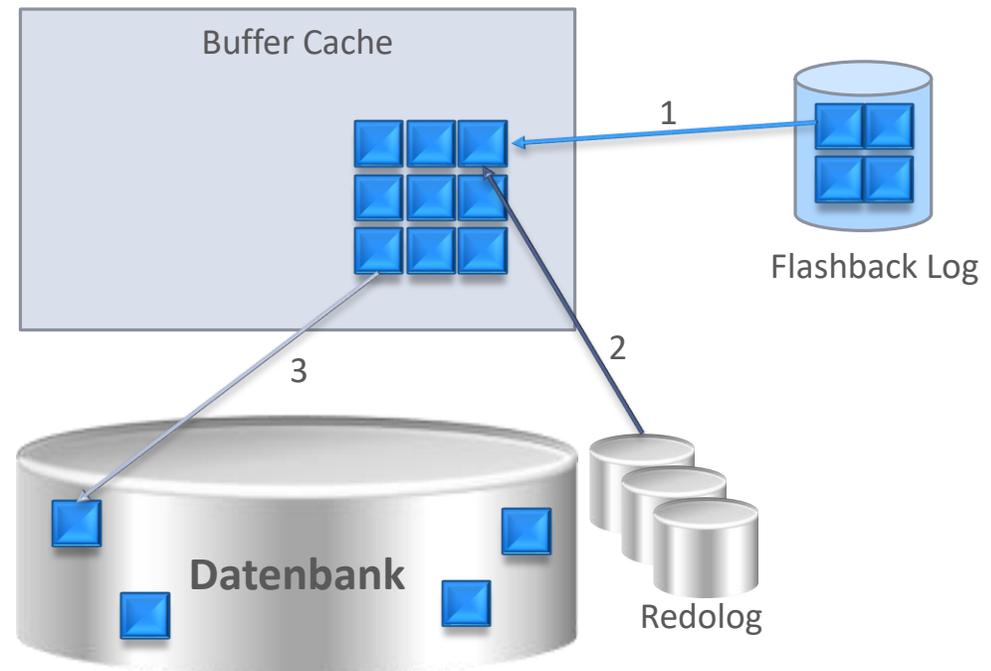
```
SQL> SELECT * FROM v$sgastat WHERE name LIKE 'flashback %';
```

POOL	NAME	BYTES	CON_ID
shared pool	flashback generation buff	4194304	1

# FLASHBACK DATABASE

```
SQL> SHUTDOWN IMMEDIATE
SQL> STARTUP MOUNT
SQL> FLASHBACK DATABASE TO <Restore Point>;
SQL> ALTER DATABASE OPEN RESETLOGS;
```

1. Flashback Log wird zurückgesichert als Datenbank Block
2. Archivelog Informationen überschreibt den Datenbank Block
3. Block wird in die Datenbank geschrieben



# Beispiel Pluggable Database

- Fehler durch Löschen einer Schemas

```
SQL> SELECT sysdate FROM dual;
SYSDATE
-----
27.10.2017 13:24:40
SQL> ALTER SESSION SET CONTAINER=suzanne;
Session altered.
SQL> DROP USER demo CASCADE;
User dropped.
SQL> CONNECT / AS SYSDBA
SQL> ALTER PLUGGABLE DATABASE suzanne CLOSE IMMEDIATE;
Pluggable database altered.
```

# Beispiel Pluggable Database

- Flashback auf bestimmten Zeitpunkt

```
SQL> FLASHBACK PLUGGABLE DATABASE suzanne TO TIMESTAMP  
      to_date('27.10.2017 13:24:00', 'DD.MM.YYYY HH24:MI:SS');
```

Flashback complete.

```
SQL> ALTER PLUGGABLE DATABASE suzanne OPEN RESETLOGS;
```

Pluggable database altered.

```
SQL> connect demo/demo@suzanne
```

Connected.

```
SQL> SELECT count(*) FROM personen;
```

```
  COUNT (*)
```

```
-----
```

```
  42000
```



# Einsatzbeispiele

# Warum Flashback Database?

- Fehlerhafter Batchprozess
- Validierung eines Programmablaufs
- Upgrade oder Patch Test
- Performancetests
- Zurücksetzen einer einzelnen Tabelle
- Zeitlicher Vergleich von zwei Datenbeständen

# Guaranteed Restore Point (GRP)

- Datenbank kann auf einen bestimmten Punkt zurückgesetzt werden
- Upgrade Test (z.B. Standby First Apply)
- Batchprozess Validierung
  - Voraussetzung: die gesamte Datenbank / PDB ist betroffen
  - GRP kann ggf. im Batch Prozess selbst gesetzt werden
  - Automatische Freigabe / Löschen nach Batchvalidierung
- GRP kann ggf. auch mehr als 24 Stunden bestehen bleiben

# Flashback Database

- Datenbank kann auf jeden beliebigen Stand zurückgesetzt werden
- Beschränkung maximal `db_retention_target` (z.B. ein Tag)
  - Kann ggf. unterschritten werden

- „In der Anwendung kann es zu Problemen kommen, dann müssen wir die Anwendung zurücksetzen können“
- Ist das die einzige Anwendung in der Datenbank?
- Ist davon ganz bestimmt keine andere Anwendung betroffen?
- Sollen wirklich alle Tabelleninhalte zurückgesetzt werden?

**NEIN!**

- Einsatz der Multitenant Database Option
  - Jede Anwendung bekommt seine eigene PDB
  - Voraussetzung: Oracle 12.2 (oder 18)
- Restore der Datenbank auf einem zweiten Server
  - Speicheranforderung beachten
  - Lizenzierung?
- Data Pump Export
  - Die ultimative Waffe gegenüber jedem Benutzerfehler

- Data Guard
  - Stoppen des Apply Prozesses
  - Öffnen der Datenbank als Snapshot Standby
    - Dabei wird ein GRP erzeugt
  - Flashback der Datenbank auf den notwendigen Zeitpunkt
  - Exportieren der notwendigen Tabellen / Schemata
  - Alternativ: erst einmal ansehen, wie die Daten zu dem Zeitpunkt aussehen

# Alternative 2

- Storage Snapshot
- Erstellen eines Snapshots aus dem Datenbank Backup zum Zeitpunkt x
- Z.B. NetApp SnapCenter
- Es wird kein zusätzlicher Speicher benötigt (C.O.W.)
- Zusätzliche Instanz auf gleichem Server  
(je nachdem, wie der Storage aufgebaut ist)

- Cloning Software
- Beispiel Actifio
- Backups werden als Image Copies in einem speziellen Storage abgelegt
  - Ein RMAN Full Backup
  - Incremental RMAN Backups forever
  - Backup der Archivelogs
- Die Datenbank wird direkt auf dem Backup (beliebiger Zeitpunkt) gestartet
  - Kann beliebig lange bestehen bleiben
  - Speicherersparnis durch Copy On Write

# ... und Data Pump?

- **Nachteile:**
  - Laufzeit kann kritisch werden (ORA-01555 Snapshot too old)
  - Benötigt ggf. viel Speicher, der nicht benutzt wird (Flashback Database auch!)
  - Einspielen / Import ist langwierig
- **Vorteile:**
  - Ausgereifte Technik
  - Exports können langfristig gespeichert werden
  - Import kann in ein anderes Schema / Tabellennamen erfolgen (Vergleich!)
  - Hohe Kompatibilität
    - siehe Dierk Lenz (2013): „*Lohnt der Upgrade von Oracle5 auf Oracle 12c?*“

- Nutzen von Pluggable Databases für die Versionierung bzw. bei Upgrades
- Speziell für Development
- Ggf. Ressourcenintensiv
- Kann auch bei Standard Edition eingesetzt werden

- Erstellen einer PDB

```
SQL> CREATE PLUGGABLE DATABASE cello  
2 ADMIN USER pdb_admin IDENTIFIED BY manager  
3 DEFAULT TABLESPACE users  
4 DATAFILE size 100M AUTOEXTEND ON NEXT 10M MAXSIZE 1000M;
```

Pluggable database created.

```
SQL> ALTER PLUGGABLE DATABASE cello OPEN;
```

Pluggable database created.

```
SQL> ALTER PLUGGABLE DATABASE cello SAVE STATE;
```

Pluggable database altered.

- Anlegen eines Services

```
SQL> ALTER SESSION SET CONTAINER=cello;
```

```
Session altered.
```

```
SQL> execute dbms_service.create_service('erfurt','erfurt');
```

```
PL/SQL procedure successfully completed.
```

```
SQL> execute dbms_service.start_service('erfurt');
```

```
PL/SQL procedure successfully completed.
```

# PDB Versionierung (3)

- ... (Nutzung der PDB)
- Erstellen eines Manifests

```
SQL> ALTER PLUGGABLE DATABASE cello CLOSE IMMEDIATE;
```

```
Pluggable database altered.
```

```
SQL> ALTER PLUGGABLE DATABASE cello UNPLUG INTO '/home/oracle/demo/cello.xml';
```

```
Pluggable database altered.
```

```
SQL> DROP PLUGGABLE DATABASE cello;
```

```
Pluggable database dropped.
```

- Erstellen einer neuen Kopie

```
SQL> CREATE PLUGGABLE DATABASE cello_18112017 USING '/home/oracle/demo/cello.xml';
```

```
Pluggable database created.
```

```
SQL> ALTER PLUGGABLE DATABASE cello_18112017 OPEN;
```

```
Pluggable database altered.
```

```
SQL> ALTER SESSION SET CONTAINER=cello_18112017;
```

```
Session altered.
```

```
SQL> execute dbms_service.start_service('erfurt');
```

```
PL/SQL procedure successfully completed.
```



# SE2 und GRP

# Refreshable PDB

- Automatische Synchronisation zwischen Quell- und Ziel-PDB
- Gedacht für Test und Entwicklung
- Refreshmodes:
  - None
  - Manual
  - Every  $n$  Minutes
- Ziel-PDB bleibt geschlossen

# Erstellen der PDB

```
SQL> CREATE PLUGGABLE DATABASE cello
      ADMIN USER pdb_admin IDENTIFIED BY manager
      DEFAULT TABLESPACE users
      DATAFILE size 100M AUTOEXTEND ON NEXT 10M MAXSIZE 1000M;

SQL> ALTER PLUGGABLE DATABASE cello OPEN;

SQL> ALTER PLUGGABLE DATABASE cello SAVE STATE;
```

# Erstellen eines Services

```
SQL> ALTER SESSION SET CONTAINER=cello;
```

```
SQL> execute dbms_service.create_service('erfurt','erfurt');
```

```
SQL> execute dbms_service.start_service('erfurt');
```

# Anlegen eines Common Users

- Wird für den Datenbank Link benötigt (Quelldatenbank)

```
SQL> CREATE USER c##johannes IDENTIFIED BY manager CONTAINER=ALL;
```

```
User created.
```

```
SQL> GRANT create session, sysoper, sysdba TO c##johannes CONTAINER=ALL;
```

```
Grant succeeded.
```

- Derzeit muss explizit sysoper Privileg vergeben werden, sonst funktioniert der Refresh nicht!

# Anlegen eines Datenbank Links

- Datenbank Link auf die CDB (nicht auf die PDB!)

```
SQL> CREATE DATABASE LINK UDO
2 CONNECT TO c##johannes IDENTIFIED BY manager
3 USING 'UDO';
```

Database link created.

- Beispiel: Refresh jede Minute

```
SQL> CREATE PLUGGABLE DATABASE cello
2 FROM cello@udo
3 REFRESH MODE MANUAL;
```

Pluggable database created.

```
SQL> show pdbs
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	CELLO	MOUNTED	

- VORSICHT: die Ziel-PDB darf nur Read-Only geöffnet werden!

```
SQL> ALTER PLUGGABLE DATABASE cello OPEN READ ONLY;
```

```
Pluggable database altered.
```

```
SQL> ALTER SESSION SET CONTAINER=cello;
```

```
Session altered.
```

```
SQL> SELECT count(*) FROM demo.personen;
```

```
  COUNT (*)  
-----  
         1000
```

# Refresh Mode überprüfen

```
SQL> SELECT pdb_name, con_id ,refresh_mode, refresh_interval  
2      FROM cdb_pdbs
```

PDB_NAME	CON_ID	REFRES	REFRESH_INTERVAL
PDB\$SEED	2	NONE	
CELLO	3	MANUAL	

- Flashback Database ist einfach
- Oracle 12.2 mit der Multitenant Database Option ist hilfreich
  - Auch bei der Standard Edition
- Data Guard physical Standby kann eine große Hilfe sein
- Die größte Herausforderung:
  - Kann die gesamte Datenbank zurückgesetzt werden?

- DOAG Schulungstag
    - Oracle 12.2 New Features (ausgebucht)
  - IT-Tage 2017
    - Oracle 12.2 New Features & Best Practices
    - Muss es unbedingt die Enterprise Edition sein?
  - DOAG Webinar Data Guard Best Practices
  - Toad User Konferenz 2018
  - DOAG SIG Database Virtualisierung
- 
- Call For Papers für die DOAG Datenbank 2018 ab dem 20. November DOAG Datenbank 2018

24. November 2017

11. bis 14. Dezember 2017

11. Dezember

14. Dezember 10:00 Uhr

12. Januar 2018 10:00 Uhr

25. Januar 2018 in Köln

07. März 2018 in Hannover

14. bis 15. Mai 2018

- Experten mit über 25 Jahren Datenbank Erfahrung
  - Firmensitz in Erftstadt bei Köln
- Spezialisten für Oracle und PostgreSQL Datenbank Administration
  - Hochverfügbarkeit
  - Optimaler Einsatz der Datenbank (z.B. Standard Edition)
  - Migrationen
  - Replikation
  - Performance Optimierung
  - Datenbank Cloning (Delphix, Actifio)
- Fernwartung
- Schulung und Workshops (Oracle, Toad, Replikation)

- E-Mail: [johannes.ahrends@carajandb.com](mailto:johannes.ahrends@carajandb.com)
- Homepage: [www.carajandb.com](http://www.carajandb.com)
- Adresse:
  - CarajanDB GmbH  
Siemensstraße 25  
50374 Erftstadt
- Telefon:
  - +49 (22 35) 1 70 91 84
  - +49 (1 70) 4 05 69 36
- Twitter: [carajandb](https://twitter.com/carajandb)
- Facebook: [johannes.ahrends](https://www.facebook.com/johannes.ahrends)
- Blog: [blog.carajandb.com](http://blog.carajandb.com)

# Weihnachtsgeschichten

- 2013 Die Multitannen Datenbank
- 2014 Der Stapelweihnachtsbaum
- 2015 Weihnachten as a Service
- 2016 Weihnachten 12.2
- 2017 ???

[blog.carajandb.com](http://blog.carajandb.com)  
ab dem 01.12.2017

