Oracle database logo should be there... but as of this page it can't be: ...Remember that you are generally not able to use Oracle logos unless you are specifically licensed or authorized to use them. ...

Aleš Zelený (zeleny.ales@gmail.com), CSPUG, CC BY-SA
Agenda

• Mind Migration
• Some terminology
• “Architecture”
• Security
• Backup and Recovery
• High Availability / Disaster recovery
• Other unordered stuff to consider
Purpose of these slides

- Understand differences
- Commons in behavior
- Displeasure prevention
- Migration guide
- RDBMS ranking
- Feature matrix
Mind Migration

- PostgreSQL is not an Oracle database
- Oracle and PostgreSQL are both superb databases, their relation is not like Red Hat and CentOS
- Do not expect equivalents for all of the Oracle RDBMS features in PostgreSQL
- Don't hesitate to be impressed by PostgreSQL broad range of data types
- PostgreSQL has extensions
Terminology

- **Architecture/concepts**
  - Cluster
  - Instance
  - Database
  - Tablespace

- **Logical**
  - Role
  - User
  - Schema
  - Tuple

- quick browse through manuals will help and don't take ages
  - Architectural Fundamentals
  - Documentation index

Yes, a cluster can simply mean a grouping of related things, but context is everything.  
Shaun M. Thomas

Wiki: A tuple is a finite ordered list of elements
Oracle and DB relation

• From installed software to database (simplified)
  - Instance is software loaded into memory working with **ONE** database (12c PDB changed that rule)

```
Oracle SW installation
ORACLE_HOME
/u01/app/oracle/product/11.2.0.4/db
```

```
Oracle SW installation
ORACLE_HOME
/u01/app/oracle/product/12.1.0.2/db
```

```
Oracle instance
ORACLE_SID
SALES
```

```
Database
SALES
```

```
Oracle instance
ORACLE_SID
EMPLOYEE
```

```
Database
EMPLOYEE
```

```
Oracle instance
ORACLE_SID
DWH
```

```
Database
DWH
```

```
Oracle instance
ORACLE_SID
INWEB
```

```
Database
INWEB
```

PostgreSQL and DB relation

- From installed software to database (simplified)
  - SW installed from RPM/APT/compiled from source

```
PostgreSQL installation 9.4
POSTGRESQL_HOME

PostgreSQL instance
Running postgres process
PGDATA
/var/lib/pqsql/9.4/data

PostgreSQL installation 9.3
POSTGRESQL_HOME
/usr/bin/postgres : /usr/lib64/libpq.so

PostgreSQL instance
Running postgres process
PGDATA
/var/lib/pqsql/9.4/data2
```

Database
- template0
- template1
- postgres

Applications databases
- INWEB
- DWH
- SALES

Internal purpose, meta data catalog
Pg & Oracle – compared

- From installed software to database (simplified)

PostgreSQL installation 9.4
POSTGRESQL_HOME

Oracle SW installation
ORACLE_HOME
/u01/app/oracle/product/12.1.0.2/db

PostgreSQL instance
Running postgres process
PGDATA
/var/lib/pgsql/9.4/data

Oracle instance
ORACLE_SID
DWH

INWEB
DWH
SALES
Database
DWH

Database
INWEB
Oracle – DB physical structure

- **Oracle RAC (Real Application Clusters)**
  - Even with RAC setup, an Oracle instance serves **ONE** database
    - Multitenant pluggable databases in version 12 breaks that rule...

- **Oracle database physical components (files)**

  - Host: first
    - Instance: MYDB1
  - Host: second
    - Instance: MYDB2
  - Host: third
    - Instance: MYDB3
  - Host: ..nth
    - Instance: MYDB<N>

  - Parameter file
  - Password file
  - Data files
  - Control files
  - Redo Log files
  - Database MYDB
  - UNDO TS
  - Archived Logs
Postgres – cluster physical structure

Not means clustering like Oracle RAC

• Postgres uses directory (might be referred by environment variable PGDATA) traditionally called **database cluster** to store all necessary data managed by Postgres instance
  
  – Which is mandatory parameter for starting Postgres instance
  – Contains configuration files, and in default setup also files for all databases residing within a particular Postgres cluster, see documentation **Database File Layout**

```
-bash-4.3$ ps -fu postgres
UID        PID  PPID  C STIME TTY          TIME CMD
postgres 30007     1  0 19:03 ?        00:00:00 /usr/bin/postgres -D /var/lib/pgsql/data -p 5433

```

**base/**
- global/
- pg_clog/
- **pg_hba.conf**
- **pg_ident.conf**
- pg_log/
- pg_multixact/
- pg_notify/
- pg_serial/
- pg_snapshots/

**pg_stat/**
- pg_stat_tmp/
- pg_subtrans/
- pg_tblspc/
- pg_twophase/
- PG_VERSION
- pg_xlog/
- **postgresql.conf**
- postmaster.opts
- postmaster.pid

**Configuration files**

**Databases directory**

```
postgres=# select oid, datname from pg_database;
          oid |      datname
---------+-------------------
        1 | template1
   12968 | template0
   12973 |    postgres
(3 rows)
```

```
-bash-4.3$ du -sh base/*
6.4M     base/1
6.4M     base/12968
6.5M     base/12973

```

```
tree base | head -5
  1
  |  12706
  |  12706_fsm
  |  12706_vm

```
Postgres – mapping to Oracle files

- Postgres similar to Oracle might create archive of online logs for PITR and other purposes
  - Instead of “Oracle Archiver” server processes, Postgres used to call any external command responsible for copying inactive online log to some other destination

```
-bash-4.3$ ps -fu postgres
UID   PID  PPID  C STIME TTY          TIME CMD
postgres 30007     1  0 19:03 ?        00:00:00 /usr/bin/postgres -D /var/lib/pgsql/data -p 5433
```

Configuration files

```
database cluster
/var/lib/pgsql/data
```

- **Database cluster**
- **Databases directory**
  - base/
  - global/
  - pg_clog/
  - pg_hba.conf
  - pg_ident.conf
  - pg_log/
  - pg_multixact/
  - pg_notify/
  - pg_serial/
  - pg_snapshots/
  - pg_stat/
  - pg_stat_tmp/
  - pg_subtrans/
  - pg_tblspc/
  - pg_twophase/
  - PG_VERSION
  - pg_xlog/
  - postgresql
  - postmaster.opts
  - postmaster.pid

```
postgres=# select oid, datname from pg_database;
     oid | datname
-------+-----------
        1 | template1
    12968 | template0
    12973 | postgres
(3 rows)
```

- **Parameter file**
- **Online Redo**
- **Parameter file**
- **Password file**
- **Control files**
- **Data files**
- **Archive command ... Archived Logs**
Tablespaces and filesystems

• Tablespaces might reside on different filesystem
  – Outage prevention
  – Data & storage tier life cycle management
    • Online active data on SSD
    • Archive data on rotating disks
    • Tablespace for temporary files – fast might be unprotected storage, no data loss risk

• Wiki page - File System Layouts
Where is UNDO tables?ace? 

- Answer: inside the data files
- Is this feature free of charge?
- No, space maintenance (vacuum) is needed to avoid table bloat.

- Postgres manages data consistency using MVCC model (Multiversion Concurrency Control)
  - Transaction isolation for each session
  - Snapshot of data visible to each session based on transaction number
  - Minimize locking contention
  - Readers never blocks writes
  - Serializable snapshot Isolation is available
Online REDO ~ WAL files

- Online REDOlogs are cyclic buffer in Oracle
  - “cleaned up” by archiver process
  - Static amount of redolog Groups each with one or more members within a redolog group
  - Log switch tuning
- WAL – Write Ahead Log files (XLOGs)
  - “cyclic buffer space” with only soft limit in size
  - File reuse – rename already archived file
  - archive_command is used called each time WAL is switched to new file (there is no “archiver” process in postgres)
WAL files

• Place them on separate filesystem
  - Up to 9.4 space requirement for XLOG filesystem
    • \( (2 + \text{checkpoint\_completion\_target}) \times \text{checkpoint\_segments} + 1 \) or \( \text{checkpoint\_segments} + \text{wal\_keep\_segments} + 1 \) files. Each segment file is normally 16 MB.
  - Starting with 9.5
    • \text{wal\_min\_size} (default 80MB ~ 5 xlogs) and \text{wal\_max\_size} (default 1GB ~ 64 xlogs)
    - Amount of WAL segments between automatic WAL checkpoint (higher values ~ potentially more data for datafiles recovery after server crash)
    - Both configurations are SOFT limit only
Archive_mode and WAL_level

• Archive_mode
  – off, on, always (archive again on streaming replica)

• wal_level
  – minimal
    • Used for crash recovery only ~ Oracle noarchivelog
  – Archive
    • Used for streaming replication ~ Oracle DataGuard
  – hot_standby
    • Used for streaming replication with read only access to replica ~ Oracle Active DataGuard
Memory

Oracle structure:

Parameters relation:

<table>
<thead>
<tr>
<th>Oracle</th>
<th>Postgres</th>
</tr>
</thead>
<tbody>
<tr>
<td>db_cache_size</td>
<td>sharedBuffers</td>
</tr>
<tr>
<td>sort_area_size (pga_aggregate_target)</td>
<td>work_mem, temp_buffers</td>
</tr>
<tr>
<td>log_buffer_size</td>
<td>wal_buffers</td>
</tr>
<tr>
<td></td>
<td>maintenace_work_mem</td>
</tr>
<tr>
<td></td>
<td>effective_cache_size</td>
</tr>
</tbody>
</table>

Postgres structure:

Nice description can be found at link: PostgreSQL 9.0 Architecture

<table>
<thead>
<tr>
<th>SHARED MEMORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>• shared buffers</td>
</tr>
<tr>
<td>• wal buffers</td>
</tr>
<tr>
<td>• CLOG buffers</td>
</tr>
<tr>
<td>• Locks space</td>
</tr>
<tr>
<td>• Other buffers</td>
</tr>
</tbody>
</table>

Per process MEMORY
• Work mem
• Maintenanace work mem
• Temp buffer
• Catalog cache
• Optimizer/executor
• Oracle database structure (simplified)
• PostgreSQL database structure (simplified)

**Logical**
- DB cluster
- Database
- DB Objects (table, index...)
- 8k pages

**Physical**
- PGDATA
  - pg_default → PGDATA/base
  - pg_global → PGDATA/global
  - pg_tblspc/

- Data files
  + free space map (_fsm)
  + visibility map (_vm)
  1GB default size limit, subsequent files are created automatically

- Operating system block

File name: pg_class.relfilenode

Cluster wide tables eg. pg_database
Default tablespace

Custom Tablespaces: symlinks to physical directories
• PostgreSQL database structure (simplified)
Architecture - connections

• Oracle
  – Process named **listener** is responsible to handle new connections
    • listener.ora (network restrictions, TCP port)
    • sqlnet.ora (protocol configuration, kerberos...)
  – Dedicated server processes per client
  – Multi-threaded server
    • Always used on Windows

• PostgreSQL
  – Master process **postgres** listens for new connections
    • pg_hba.conf (user/database/network restrictions)
    • postgresql.conf (TCP port, kerberos, RDBMS configuration...)
  – Dedicated server only
    • Shared memory and semaphores are used for inter process synchronization
  – Connection pooling by other products
    • PgBouncer
    • pgpool-II
Architecture notes

• Oracle
  - Decided that RDBMS is right and only place to manage database buffers
  - Promotes its ASM to have a direct control on file management (ASM is kind of LVM dedicated to Oracle)

• PostgreSQL
  - Relies on (believes to) OS file cache management
  - Do not re-implement features already implemented in OS, thus it use file system to store its data files (no RAW device support)
Security observations I

• Oracle has users and roles
  - Users and Roles are defined on DB level (not applies for PDB)
  - Users and Roles are different entities
• Postgres has roles only
  - Some roles might be granted “with login” permission
• Oracle schema consist from a single user objects (schema = user)
  - Schema is not an object, so it can't be granted
• Postgres schema is a grantable name-space object
  - Ownership and usage on schema might be granted to roles
  - Objects owned by different roles (users) might reside within a single schema
  - Public schema might (and should) be dropped
Security observations

SW installation:
/oracle/product/12.1.0.2/db_1
OS access control to files

user1

\[ X - \text{Doesn't work} \]

user3

Schema: user3

TableA

TableB

FunctionC

RoleOne

RoleTwo

user2

TableA

TableB

FunctionC

Schema: user2

\[ X - \text{Doesn't work} \]
Security observations II

- Oracle tablespace always belongs to a database
  - quotas might be used to limit tablespace usage by users

- Postgres tablespace is defined at cluster level
  - "create" on TS might be granted to a role
  - TS ownership to a role might be defined
  - There are no space usage quotas on tablespace, check FS free space

- Oracle database contains users defined inside DB, there is no database ownership concept
  - Grant scope is always within a database (PDB global users exception exists)

- Postgres database might be owned by a specific role
  - One role might have granted access on objects from multiple databases
  - Role attributes possible in scope of database – alter role XXX search_path = YYY,ZZZ in MY_DATABASE
Security observations III

• Oracle distinguish
  – System privileges (create table..., select any ... )
  – Object privileges (grant select on ... )

• Postgres does not have such strong difference
  – Login permission is cluster wide kind of “system” privilege
  – Mostly all privileges are related to some objects including database object itself
    • Grant connect on database myDB
    • Grant usage on ...
    • Grant create on ...
Security observations IV

- **Oracle Advanced Security**
  - Transparent Data Encryption
  - Kerberos (MS AD integration) is available without Advanced security as of 12.1 release, applies to older releases
  - Many other security features (VPD, RLS...)

- **Postgres**
  - SSO available
  - *Row Security Policies* are available with 9.5 release
  - TDE is not available
  - Encryption is covered by separate module *pgcrypto*
Security observations V

- **Oracle remote access control**
  - IP address level: sqlnet.ora
    - tcp.validnode_checking = yes
    - tcp.invited_nodes = (hostname1, hostname2)
    - tcp.excluded_nodes = (192.168.10.3)
  - username password and create session is evaluated as next step

- **Postgres**
  - `pg_hba.conf` File
    - username/role_membership, database name, source IP address and authentication method is evaluated prior password validation
    - Password is evaluated as next step
Security observations VI

- **Oracle [public] synonyms**
  - Synonyms are used to reference another user (schema) objects
  - Might be defined as public – accessible to all users

- **Postgres**
  - `search_path` session environment is used to define scope of visible objects, used similar to PATH in OS
    - Might be defined at cluster level
    - Users might have specified different search path values in particular databases

```
ALTER ROLE { roleSpecification | ALL } [ IN DATABASE database_name ] SET configuration_parameter { TO | = } { value | DEFAULT }
```
Security features...

- **ALL macro in grant commands**
  - Expands to all at time of execution existing objects satisfying grant scope criteria
    - Grant execute on **ALL functions** in schema my_schema to ...

- **Alter default privileges**
  - Does not affect existing objects, applied to newly created ones
  - Doc: `ALTER DEFAULT PRIVILEGES`  
    `ALTER DEFAULT PRIVILEGES`  
    `[ FOR { ROLE | USER } target_role [, ...] ]`  
    `[ IN SCHEMA schema_name [, ...] ]`  
    `abbreviated_grant_or_revoke`
Backup … and recovery

• Database [full or partial] dump
  – Oracle exp/imp, expdp/impdp
  – Postgres
    • pg_dump / pg_restore
      – “directory” format supports parallel dumps
    • pg_dumpall (use it for cluster globals only)
      – Load dump by call to psql
    • Thanks to MVCC, there is no “ORA-1555” risk during dump
      – For sure, the backup is consistent even if the database is used during the dump
Binary backups and recovery

- Offline! Works for Oracle, Postgres…
- Online Oracle database backups
  - Manual
    - Alter database (tablespace) begin backup, Copy corresponding datafiles, alter database (tablespace) end backup, store archived redologs needed for recovery
  - Or use Oracle RMAN utility
- Online Postgres cluster backup
  - Backup Control Functions
    - pg_start_backup(), pg_stop_backup(), same as above for Oracle [no TS level available]
  - pg_basebackup
    - Handle calls to backup control functions and might produce copy of postgres cluster or tar archive with the backup. Some features are available like tablespace mapping for convenient backup procedure/higher flexibility
Binary backups and recovery

- **pgBarman**
  - Some features similar to oracle RMAN
    - Recovery window / # of copies
    - Stores archived WALs together with Barman backups
    - Backup reports
    - Does not use “rman catalog”, backed up files with some barman metadata files are enough
    - Single backup might be archived to tape (tape integration is not part of pgBarman) – it disappears from backup reports, once retrieved from tape, pgBarman can use the backup again

- **pgBackRest**
  - More complicated configuration than Barman, incremental backups seems to be implemented slightly better
HA & DR

- OS clusterware (RHEL Pace Maker, PowerHA…)
  - Simply works

- There is no usable technology like Oracle RAC for PostgreSQL server
  - Sharding (Postgres XL) is not about sharing data files between nodes
  - Oracle 12.2 seems to provide some support for sharding
Oracle replication

• Oracle DataGuard
  – Log shipping (log_archive_dest_n) by archiver
    • ARCHIVE_LAG_TARGET
  – Redo transmit by LGWR
    • ASYNC
    • SYNC
    • Delayed recovery - DELAY=minutes attribute of the LOG_ARCHIVE_DEST_n
  – Logical standby
  – Active Data Guard
  – Golden Gate
RDBMS replication

• Postgres
  – Log-Shipping Standby Servers
    • archive_timeout
  – Streaming Replication
    • ASYNC (default)
    • SYNC - Synchronous Replication
    • Standby Server Settings recovery_min_apply_delay available from 9.4
  – Logical Standby
    • Slony, Bucardo, logical decoding framework
  – Hot Standby ( read only accessible standby )
  – BDR provides asynchronous multi-master logical replication.
Others... I

- `psql` command line client
  - Comfortable interface, but be aware of default AUTOCOMMIT behavior
- **CZ fast reference** by Pavel Stěhule
- `--data-checksums` initdb option
  - Page check-sums are calculated for all object in all databases in cluster
  - Use `pgbench` to verify performance impact
  - Checksum is calculated on page read
  - Backup operate at file level, checksums are not calculated during backup
• **Oracle dual table**
  - `select function() from dual;`
  - SQL Loader, External tables
  - `db links`

• **PostgreSQL**
  - `select function(); select 5/8;`
  - `copy command (client side, server side), file_fdw for CSV files, format compatible with COPY command required`
  - Foreign Data Wrappers for many kinds of data sources, including Oracle database
Others... III

- **Porting from Oracle PL/SQL**
  - Oracle / Postgres – often similar, not always the same
    - ORA: trunc(date_variable, format)
    - PG: date_trunc('field', source)
- Pipelined functions are not implemented
- Group by can use column alias in PostgreSQL

```sql
open2300db=> select date_trunc('hour', rec_datetime) as record_time,
    round(avg(temp_out), 2) as avg_temp,
    max(wind_speed_max) as max_wind_max
from open2300.weather where
    rec_datetime > now() - interval '3 hour'
group by record_time
order by record_time desc;

<table>
<thead>
<tr>
<th>record_time</th>
<th>avg_temp</th>
<th>max_wind_max</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016-02-15 22:00:00+01</td>
<td>3.04</td>
<td>2.4</td>
</tr>
<tr>
<td>2016-02-15 21:00:00+01</td>
<td>3.23</td>
<td>2.5</td>
</tr>
<tr>
<td>2016-02-15 20:00:00+01</td>
<td>3.66</td>
<td>2.3</td>
</tr>
<tr>
<td>2016-02-15 19:00:00+01</td>
<td>4.11</td>
<td>2.6</td>
</tr>
</tbody>
</table>
```
Postgres partitioning is implemented on top of inheritance feature

- Declarative partitioning like in Oracle is not available
  - Some basic development for 9.6

Constraint on child tables

Trigger on master table

- Static IF... requires trigger compilation if new child partition tables are added
  - Trigger builds dynamic SQL – more overhead

No global indexes on partitioned tables
Others… get table filename

- Bonus link:
  How to find out which PostgreSQL table a file on disk corresponds to

- Q & A /* end of slides */