

PostgreSQL

Eine Oracle Alternative?!

Johannes Ahrends (CarajanDB GmbH)

- 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)
 - Singen (überall)



Warum PostgreSQL?

- Oracle ist teuer
- Schwierige Lizenzpolitik bezüglich Vmware
- Standard Edition nur mit Einschränkungen nutzbar
 - Hochverfügbarkeit
 - Keine AD Anbindung
- Druck auf Oracle aufbauen
- Wechsel in die Cloud (AWS / Azure)
- Containerisierung

Welches PostgreSQL?

- PostgreSQL Community Version
 - Open Source
 - Keine Kosten
 - Kein Vendor Lock
- Enterprise DB Advanced Server
 - Basiert auf PostgreSQL
 - Wird regelmäßig aktualisiert
 - Oracle Kompatibilität
 - 7x24 Support
- Aurora
 - PostgreSQL Derivat von Amazon
 - Nur in der AWS Cloud
 - Basiert max. auf PostgreSQL Version 10.7 (Stand Okt. 2019)

- Fünf ausgesuchte Spezialisten für folgende Aufgaben:
 - Koordination der Releases
 - Weitergabe/Kontrolle vertraulicher Informationen
 - Definition der strategischen Ausrichtung
 - Freigabe von Infrastrukturen, etc.
 - Wahrung der Disziplin
 - Schiedsrichter bei kritischen Entscheidungen
- Es dürfen nicht mehr als zwei Personen im selben Unternehmen tätig sein

PostgreSQL Core Team Member

- Peter Eisentraut
 - peter.eisentraut at 2ndquadrant.com
 - 2ndQuadrant
 - Dresden, Germany
- Magnus Hagander
 - magnus at hagander.net
 - Redpill Linpro
 - Stockholm, Sweden
- Tom Lane
 - tgl at sss.pgh.pa.us
 - Crunchy Data
 - Pittsburgh, Pennsylvania, USA
- Bruce Momjian
 - bruce at momjian.us
 - EnterpriseDB
 - Philadelphia, Pennsylvania, USA
- Dave Page
 - dpage at pgadmin.org
 - EnterpriseDB
 - Oxfordshire, United Kingdom

Major Contributors

- zur Zeit 34 Personen
- Stellen die Weichen für die Postgres Weiterentwicklung

- Basis:
 - Ingres „Interactive Graphics Retrieval System“ (1970er Jahre)
- POSTGRES Projekt (University of California in Berkeley)
 - Version 1: erste offiziell verfügbare Version (1989, intern seit 1987)
 - Version 4.2: Ende des POSTGRES Projekts (1993)
- Postgres95 (1994)
 - Open Source
 - Neuer SQL Sprachinterpreter für POSTGRES (SQL statt PostQUEL)
 - Command Line Interface „psql“
 - Komplett in ANSI C geschrieben
- PostgreSQL (1996)
 - Neuer Name (statt Postgres95)
 - Version 6.0 (vorherige Nummern wurden übersprungen)

Aktuelle Versionen

- 9.4.26
- 9.5.21
- 9.6.17
- 10.12
- 11.7
- 12.2

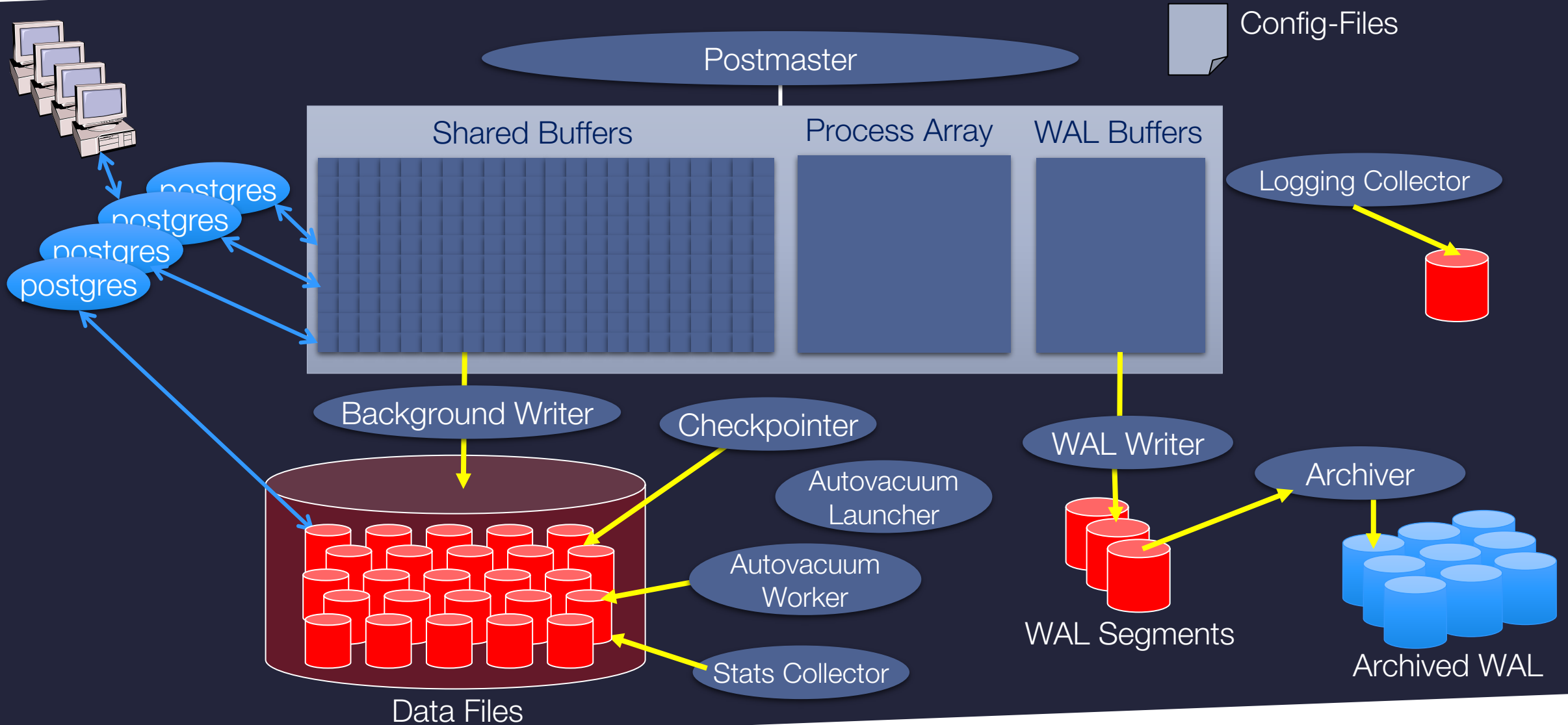
Stand März 2020

- Sehr ähnlich aufgebaut
 - PostgreSQL Datenbank-Cluster → Oracle Datenbank
 - Multi Version Consistency Check → Oracle Consistent Read Modell
 - Write Ahead Logs → Oracle Redo Logs
 - Archiving → Archiver
 - Online Backup → Begin / End Backup
 - psql → SQL*Plus

	Limit
Maximale Datenbankgröße	Unbegrenzt
Maximale Tabellengröße	32 TB
Maximale Spaltengröße	1,6 TB
Maximale Feldgröße	1 GB
Maximale Anzahl Zeilen pro Tabelle	Unbegrenzt
Maximale Anzahl Spalten pro Tabelle	250 bis 1600 (je nach Datentyp)
Maximale Anzahl Indizes pro Tabelle	Unbegrenzt

- Ausschließlich Prozesse, keine Threads
- Postmaster ist der Supervisor
 - Entspricht dem Oracle smon, aber auch pmon
 - Postmaster ist auch listener
 - Zusätzliche Hintergrundprozesse werden bei Bedarf gestartet
- Jeder Userprozess erzeugt einen Server/Backend Prozess

Architekturkomponenten

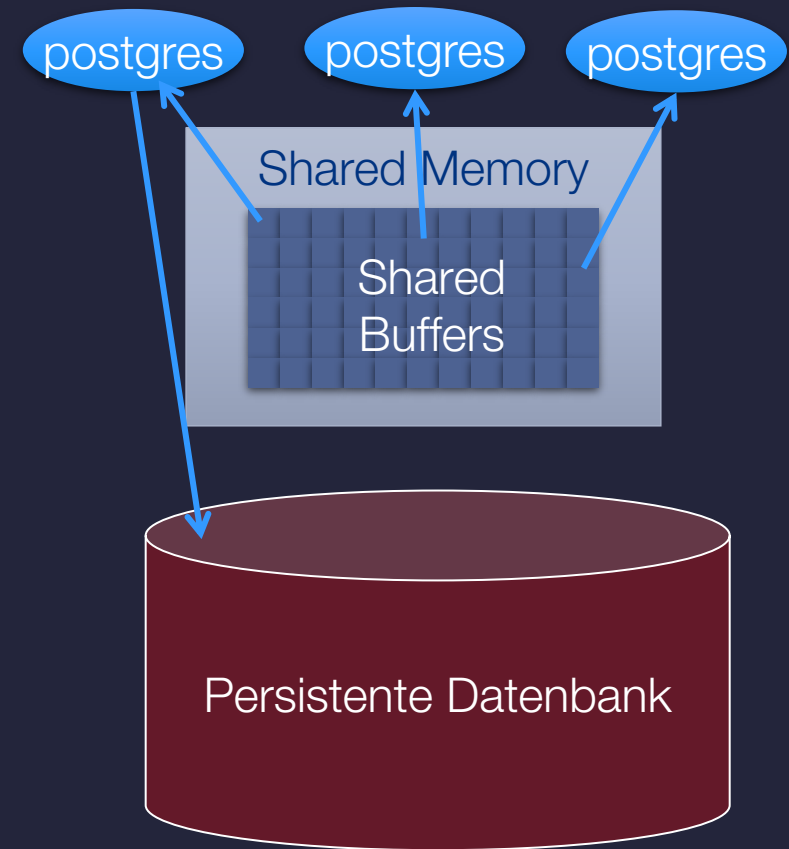


- Background Writer
 - Schreibt „Dirty“ Blocks in die zugehörigen Data Files
- WAL Writer
 - Schreibt Write Ahead Logs in die WAL Segmente
- Checkpointer Prozess
 - Initiiert Checkpoints basierend auf die Konfigurationseinstellungen
- Autovacuum Launcher
 - Startet Autovacuum Worker Prozesse
- Autovacuum Workers
 - Führen das Vacuum aus

- Logging Collector
 - Schreibt Logginginformationen in SYSLOG, EVENTLOG oder eigene Logfiles
- Stats Collector
 - Sammelt Statistikinformationen für Relationen (Segmente) und Blöcke
- Archiver
 - Archiviert die Write-Ahead Log Files

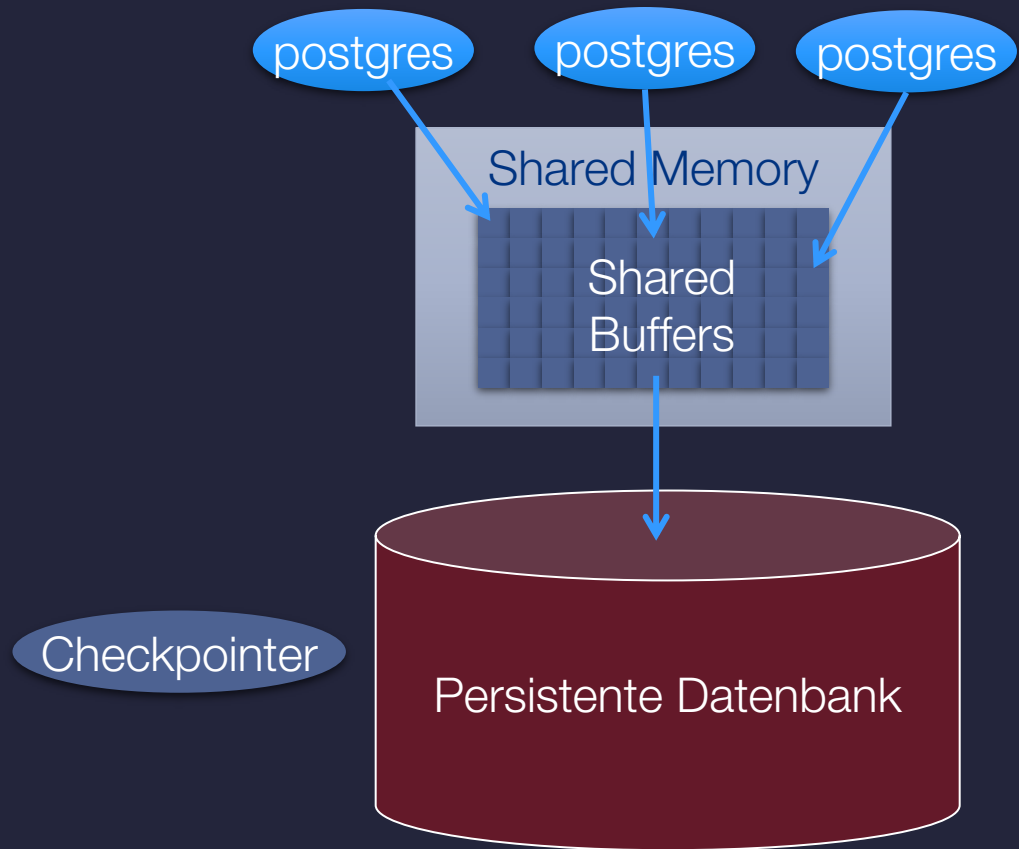
- Postmaster ist der Masterprozess
 - Name: postgres
- Horcht auf genau einem Port
 - Standard 5432 (Postgres) bzw. 5444 (EnterpriseDB)
- Forkt / Spawned einen Serverprozess (Backend Process)
 - Name: postgres
- Reicht den Client an den Serverprozess weiter
 - Kommunikation über Shared Memory und Semaphore
 - Authentifizierung und Autorisierung

- PostgreSQL Buffer Cache
 - Reduziert Diskzugriffe
 - Einmal lesen
 - Vielfach nutzen



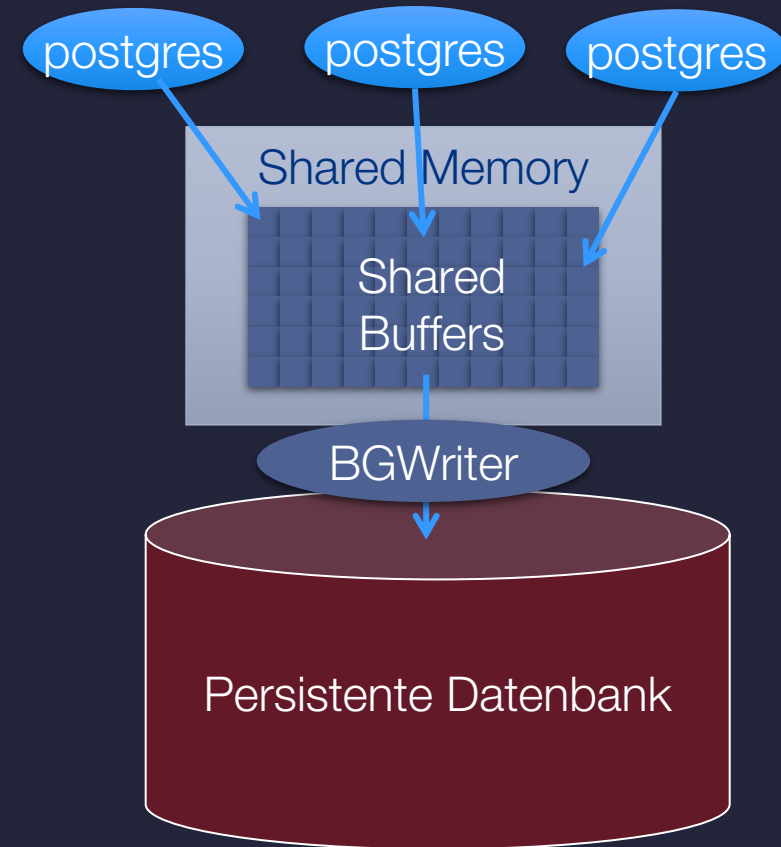
Disk Write Buffer

- Blöcke werden nur bei Bedarf geschrieben
 - Platz für neue Blöcke im Cache
 - Checkpoint



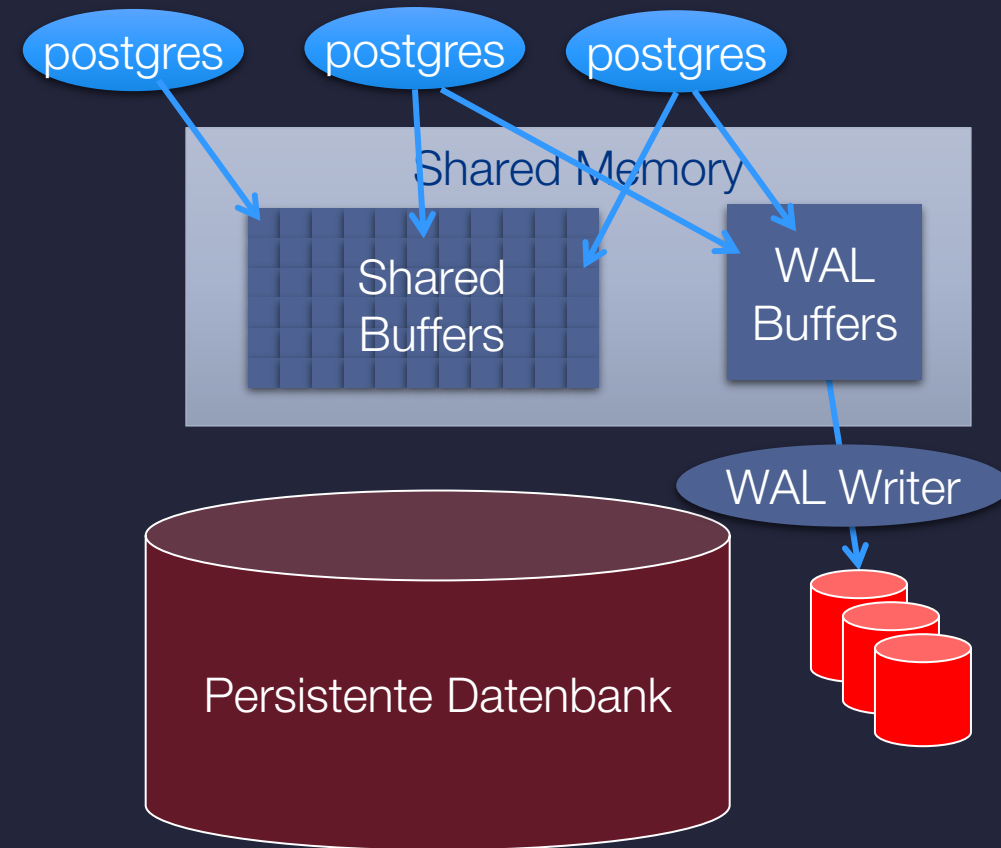
Background Writer

- Scannt permanent den Buffer
- Schreibt geänderte (Dirty) Buffer auf die Disk



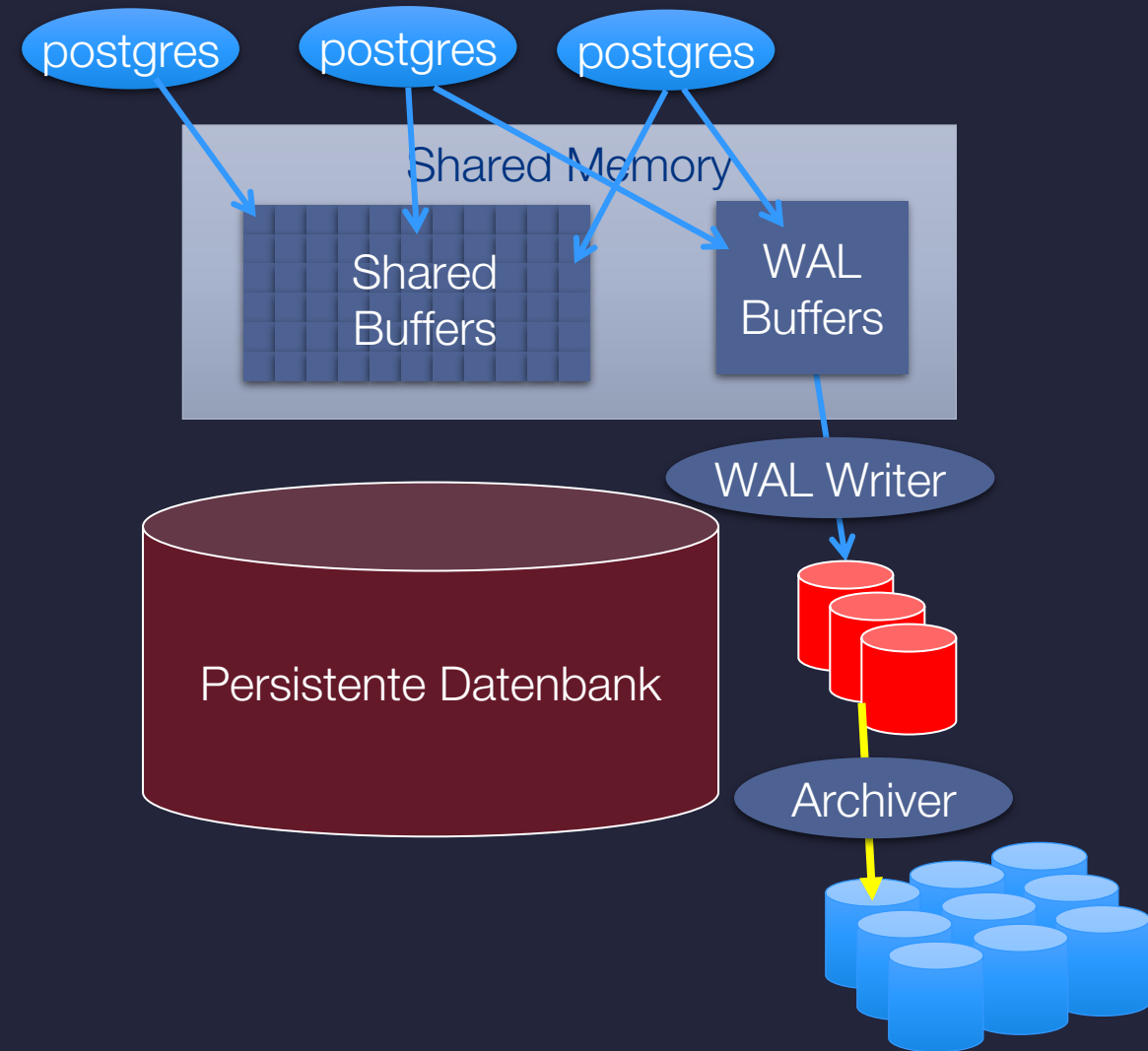
Write Ahead Logging (WAL)

- Server / Backend Prozesse schreiben in WAL Buffer
- WAL Writer schreibt WAL Buffer in WAL Segmente
 - Beim Commit
 - Wenn der Buffer voll ist
 - Group Commit möglich



Transaction Log Archiving

- Kopiert die WAL Segmente (pg_xlog) Dateien an eine andere Stelle



- Sammlung von Datenbanken verwaltet von einer Server Instanz
- Ein Data Verzeichnis pro Cluster (PGDATA)
 - Links / Tablespaces können auf andere Verzeichnisse verweisen
- Ein TCP Port
- Ein Set an Server Prozessen
- Einen Postmaster
- Mehrere Datenbanken



Installation

- Postgres Repository laden

```
yum -y install https://download.postgresql.org/pub/repos/yum/11/redhat/rhel-7-x86_64/pgdg-oraclelinux11-11-2.noarch.rpm
```

- Aktuelle Version installieren

```
yum -y install postgresql11          # nur Client  
yum -y install postgresql11-server # Server
```

- Installationsverzeichnis: /usr/pgsql-11

```
[root@paert ~]# ls -l /usr/pgsql-11  
total 12  
drwxr-xr-x. 2 root root 4096 Jan 25 12:49 bin  
drwxr-xr-x. 3 root root 4096 Jan 25 12:49 lib  
drwxr-xr-x. 7 root root 4096 Jan 25 12:49 share
```


- Software Installation
 - `/opt/...`
 - `/usr/edb/as11/bin` (EnterpriseDB Version 11)
 - Links in `/usr/bin` (Nur EDB-Postgres)

Datenbank-Cluster

Erstellen eines Clusters

- Verzeichnis auswählen
 - Schreibberechtigung für `postgres` bzw. `enterprisedb`
 - Variable `PGDATA`
- Ausführen des Befehls „`initdb`“
 - Erstellt das Verzeichnis
 - „data“ kann auch manuell erstellt werden, muss aber dem Super User (`enterprisedb` bzw. `postgres`) gehören

- Beliebiges Verzeichnis, muss „postgres“ bzw. „enterprisedb“ gehören
 - Permission 700
 - Variable PG_DATA
 - Basisverzeichnis für alle Datenbank-Dateien

```
[enterprisedb@paert ~]$ initdb -D /u02/pgdata/arvo
The files belonging to this database system will be owned by user "enterprisedb".
This user must also own the server process.
```

```
The database cluster will be initialized with locale "en_US.UTF-8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".
```

```
Data page checksums are disabled.
```

```
creating directory /u02/pgdata/arvo ... ok
creating subdirectories ... ok
```

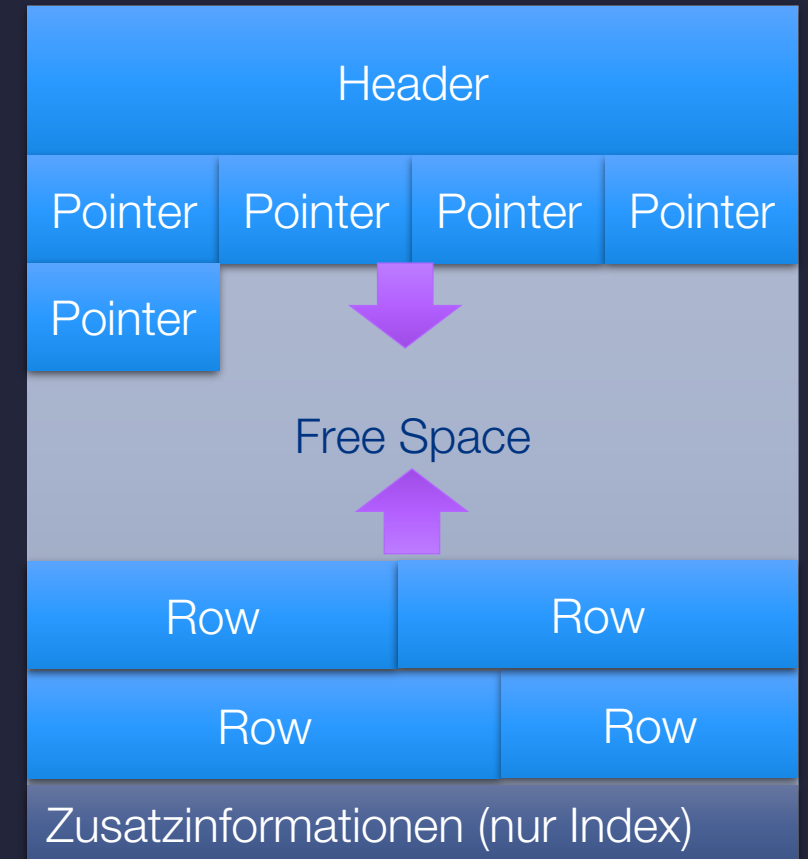
```
...
```

```
WARNING: enabling "trust" authentication for local connections
You can change this by editing pg_hba.conf or using the option -A, or
--auth-local and --auth-host, the next time you run initdb.
```

```
Success. You can now start the database server using:
```

```
pg_ctl -D /u02/pgdata/arvo -l logfile start
```

- Immer 8 kByte
- Page Header (24 Byte)
 - Generelle Informationen über den Block
 - Pointer auf den Freespace Bereich
- Row / Index Pointer (4 Byte)
 - Array mit Offset und Längenangabe als Verweis auf den Datensatz bzw. Indexeintrag
- Free Space
 - Freier Platz für neue Sätze / Einträge
- Row / Index Eintrag
- Zusatzinformationen (nur bei Indizes)
 - Informationen über den Indexzugriff





Tablespaces

- Daten werden logisch in Tablespaces verwaltet physikalisch aber in Datafiles
- Tablespaces
 - Gehören zu genau einem Datenbank Cluster
 - Beinhalten mehrere Data Files
 - Können von mehreren Datenbanken genutzt werden
- Data Files
 - Gehören zu genau einem Tablespace
 - Speichern Daten von Datenbank Objekten
 - Können nicht von mehreren Objekten geteilt werden
 - Ein Objekt hat ein oder mehrere Data Files

Warum Tablespaces?

- Verteilung von Objekten auf mehrere Disks / Filesysteme
 - Tabelle und Index
 - Schemata
 - Partitionierung
 - Disk Layout (SSD vs. HDD)
- Symbolischer Link im Database Cluster (`pg_tblspc`)

Vordefinierte Tablespaces

- `pg_global`
 - Referenziert auf `PGDATA/global` Verzeichnis
 - Data Dictionary (Cluster Wide Tables)
 - Systemkatalog
- `pg_default`
 - Referenziert auf `PGDATA/base` Verzeichnis
 - Default Tablespace für Benutzerobjekte

- Jeder Tablespace hat einen Symbolic Link in PGDATA/pg_tblspc
- Symbolic Link Name ist OID des Tablespaces
- Tablespace enthält Unterverzeichnis mit der PostgreSQL Version des Clusters
- Jede Datenbank in einem Tablespace hat ein eigenes Subdirectory
- Erstellen mit CREATE TABLESPACE

```
CREATE TABLESPACE <tablespacename> [ OWNER <username> ] LOCATION '<Verzeichnis>'
```

Security

PostgreSQL Security Level



- Host Based Access Control File
- Liegt im Database Cluster (PGDATA) Verzeichnis
- Wird beim Startup gelesen
 - Jede Änderung erfordert einen Reload
 - Top to Bottom (d.h. der erste gültige Werte wird genommen)
- Jeweils ein Record pro Zeile bestehend aus
 - Connection Typ
 - Datenbankname
 - Username
 - Client IP (IPv4 o. IPv6)
 - Authentifizierungsmethoden



Enterprise DB Advanced Server

- 7 x 24 x 52 Support
- Tools:
 - Backup und Recovery Tool (BART)
 - Postgres Enterprise Manager (PEM)
 - Migration Toolkit
 - Replication Server
 - Failover Manager (Standby Datenbank)
- Oracle Kompatibilität
 - DATE, NUMBER, VARCHAR2, ...
 - Oracle Packages und Functions

Backup and Recovery Tool (BART)

- Backup von EDB AS und PostgreSQL Datenbanken
 - Full
 - Block Level Incremental
- Einfaches Point in Time Recovery
 - Transaction ID
 - Timestamp
- Backup Monitor
- Gemeinsamer Katalog
- Retention Policies
- Backup Verification (Checksum)

- CREATE PROFILE
 - FAILED_LOGIN_ATTEMPTS UNLIMITED
 - PASSWORD_LOCK_TIME UNLIMITED
 - PASSWORD_LIFE_TIME UNLIMITED
 - PASSWORD_GRACE_TIME UNLIMITED
 - PASSWORD_REUSE_TIME UNLIMITED
 - PASSWORD_REUSE_MAX UNLIMITED
 - PASSWORD_VERIFY_FUNCTION NULL
 - PASSWORD_ALLOW_HASHED TRUE

- Hints
- edb_stmt_level_tx
- Synonyme
- Hierarchische Queries (Connect By prior)
- Data Redaction

- Verwendet Oracle OCI Library
- Zugriff aus AS auf Oracle Datenbanken
 - SELECT
 - INSERT
 - UPDATE
 - DELETE

- Setup Parameter in postgresql.conf
 - `edb_audit` [xml | csv | off]
 - `edb_audit_directory`
 - `edb_audit_filename`
 - `edb_audit_rotation_day`, `edb_audit_rotation_size`,
`edb_audit_rotation_seconds`
 - `edb_audit_destination` [`edb_audit_directory` | `syslog`]

- Experten mit über 30 Jahren Datenbank Erfahrung
- Spezialisten für
 - Datenbank Administration (Oracle und PostgreSQL)
 - Hochverfügbarkeit (RAC, Data Guard, Replication, etc.)
 - Migrationen (Unicode, PostgreSQL)
 - Performance Optimierung
 - Monitoring (OEM, Foglight, CheckMK, PEM)
- Fernwartung
- Schulung und Workshops
 - PostgreSQL
 - Oracle Multitenant
 - Toad



- E-Mail: johannes.ahrends@carajandb.com
- Homepage: 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
- Facebook: johannes.ahrends
- Blogs:
 - blog.carajandb.com
 - www.toadworld.com