2ndQuadrant
Professional PostgreSQL

Daten auf mehreren Systemen
Swiss Postgres Conference
Juni 2016

Harald Armin Massa

Alle Daten in einer Datenbank, auf einem Computer

- Eine Quelle der Wahrheit
- Viele Werkzeuge für Korrektheit
- Alle Daten können integrierend genutzt und abgefragt werden
- Geringstmögliche Komplexität

Warum überhaupt mehr als eine Datenbank, und gar auf mehr als einem Computer?

Eigene Erfahrungshistorie

- CIO bei Vertriebsgesellschaft
  - Steuerung Entwicklung von CRM / Verwaltungssystem
  - Weiterentwicklung
  - Migration FoxPro → PostgreSQL + PHP
- Eigene Firmen seit 1999, diverse Tools rund um Datenbanken
- „großer Wurf“ Browserbasierte Checklisten-Software
- Seit 2010 Partner bei 2ndQuadrant

Ziel

- Überblick der Gründe, warum Daten auf mehr als einer Maschine sein sollen / müssen
- Überblick einiger Techniken dazu
  - Einsatzbereiche
  - Kontraindikationen
  - Nebenwirkungen
- Anregungen, die richtigen Fragen zu stellen

Methodik

Lehrgeschichte:

Handtuchvertrieb Europa (HVE), gegründet in Hamburg (Deutschland)

HVVW 1.0 (Handtuchvertriebverwaltung)

Stufe 1

- HVVW 1.0 Mehrplatzanwendung
- Zentrale Datenbank
- Alles flutscht
- Angebote, Lieferscheine, Rechnungen
- Betriebswirtschaftliche Auswertungen

WUMMS.

Server kaputt.

Wunsch: Betrieb soll auch
in solchen Fällen weitergehen

Schlechte Idee

- 1 Festplatte, 2 Computer
- Storage switch over

- weniger Hardwareaufwand
- Single point of failure
- Bei Datenbank PostgreSQL:
  - Mehr eine Frage wann, nicht ob Datenbank kaputt geht

Low level Daten Duplikation

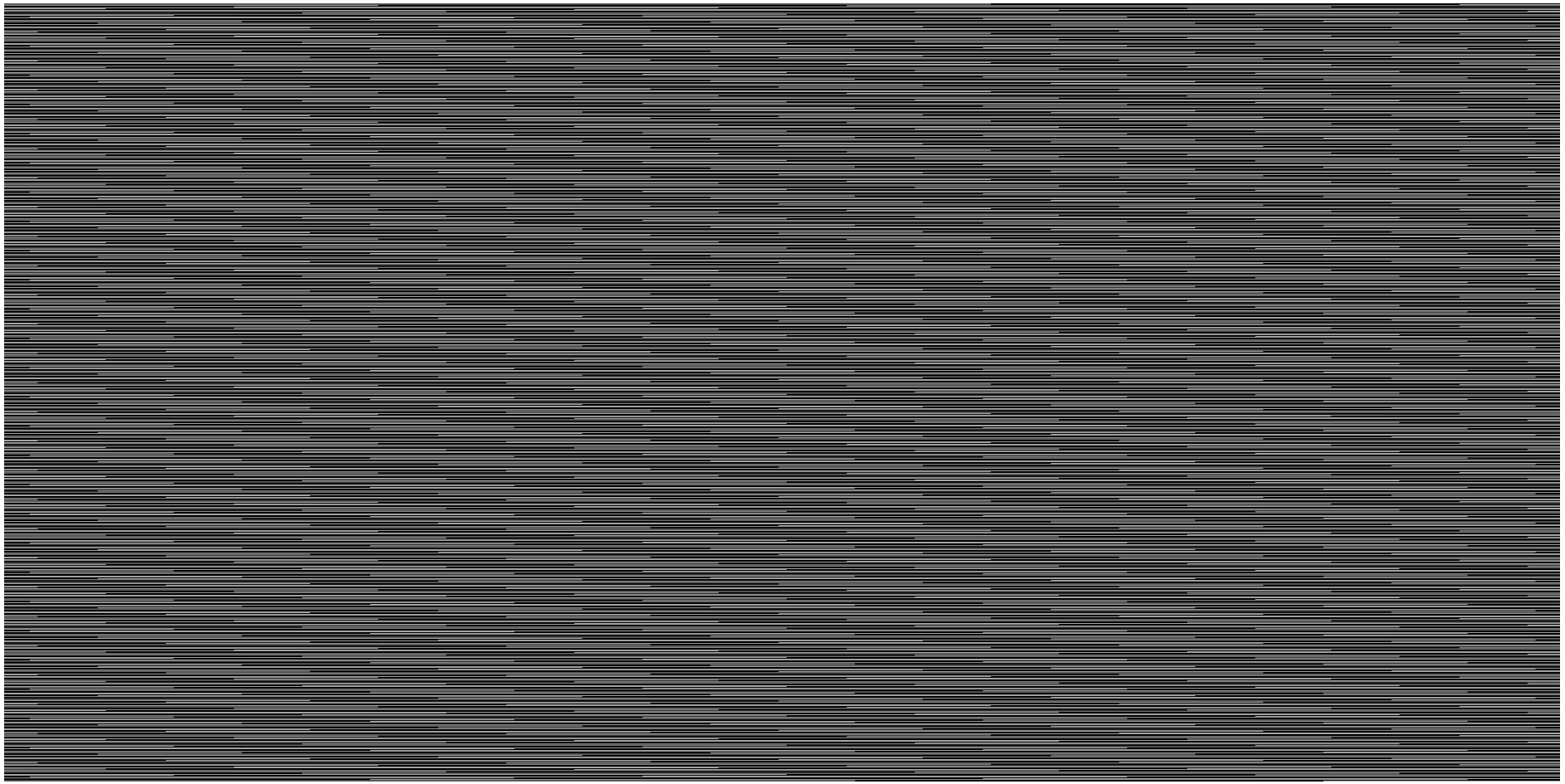
- Betriebssystem
- Virtualisierungsebene
- Storage

- Oft proprietär (virtualisierung, Storage)
- Sehr spezieller Skillset

Database Level Duplication

- Replikation
  - Trigger based
  - WAL shipping
    - Replication
    - PITR / Disaster Recovery
  - Streaming Replication
    - Sync
    - Async

Trigger basierend

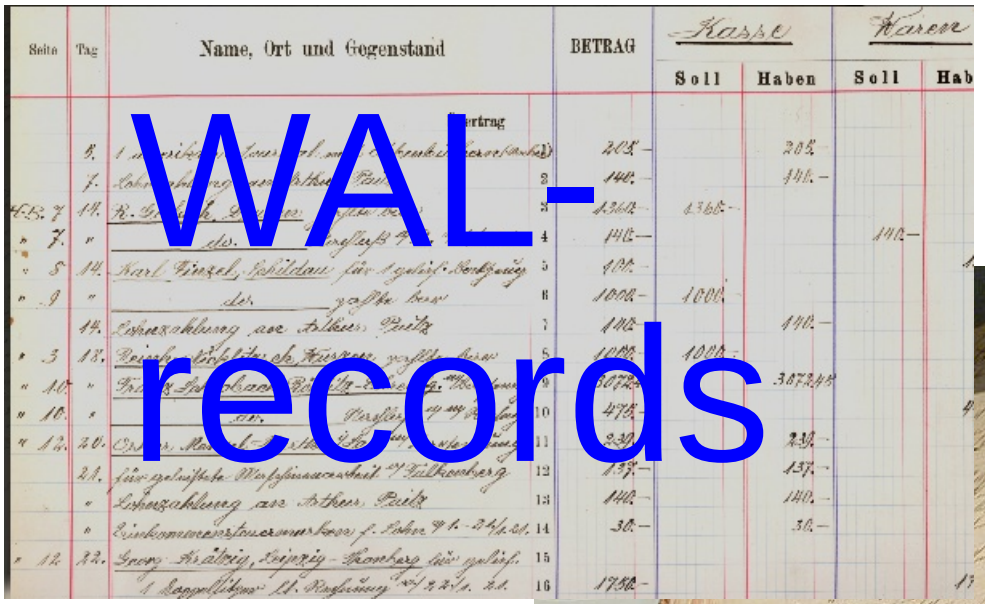


Trigger basierend

- Data werden 4x geschrieben
  - WAL, Datafile, QueueTable, WAL(QT)
- Schemaänderungen = Schmerzen
- Kosten der Trigger
- Kosten des Parsens
- Zwischen unterschiedlichen Versionen von PostgreSQL möglich
- (Sogar Other → PostgreSQL ist möglich)
- Replikation einzelner Tabellen möglich

2ndQuadrant
Professional PostgreSQL

WAL basierte Replikation



WAL-records

table files

1340 in Genua erfunden

WAL basierte Duplikation

- Gesamter Cluster
- Schemaänderungen schmerzfrei
- wenig Last auf Master
- Kein Parsingoverhead
- Slave ist Read Only
- Sichern des WAL erlaubt PITR und DR

Datenbankbasierte Duplikation

- Wartungen und Upgrades
- Ausfallsicherheit – Backup Maschine vorhanden
- Erhöhte Komplexität gegenüber SingleServer
  - More stuff to break

Regen + Hitze

- Mehr Handtücher werden gekauft
- HVVW 2.0
- Jetzt noch mehr Reportmöglichkeiten

- → mehr Datenbanklast

Technik 1

- Readonly Server für Reports
  - Per Trigger
  - Per Streaming Replication
- Trigger
  - Decoupling
  - Kodier, dekodier, Schreiblast
- Streaming Replication
  - dichter gekoppelt: vacuum

Technik 2

- ETL nach Analyse-Datenbank
- ETL nach Analyse-Tool (Excel, Qlikview…)

- Andere Schema möglich
- Preprocessing
- Aktualität?
- Belastung bei Extrakt
  - Full Table Read gegen Cache

Analyse-Datenbank

- Why not PostgreSQL?
  - 200 Milliarden Records summieren
    = 200 Milliarden Records lesen
- parallele Aggregate als Addons verfügbar
- Parallel Read in Arbeit
  - Beides bestensfalls multiple Prozessoren
  - Immer nur 1 Storage

Analyse DB

- Viele Bytes lesen → evtl. mehr als ein Storage
- Achtung, aktuelle Geschwindigkeiten
  - Viele PCIe-SSDs: 2,5 GBytes / sec lesen (06-2016)
  - Einzelne 5GBytes / sec lesen
  - 1000€ / TB Größenordnung

Analyse DB – you really need more

- PostgresXL
- Basiert auf PostgreSQL, folgt PostgreSQL eng
- Selbes SQL
- Vielen Computer erarbeiten Teilergebnisse
- Koordinator führt Ergebnisse zusammen

Nutzen und Nebenwirkungen

- Gut:
  - Bandbreite vieler Storages
  - Rechenleistung vieler Prozessoren
  - Cache-Wirkung vieler Hauptsspeicher
- Preis
  - Logik des Verteilens & Zusammenführens
  - Rechenzeit des Zusammenführens
  - Netzwerkbandbreite + Latenz

Wir müssen ins Internet

- Handelspartner fordern Online-Information
- Kunden wollen Auftragsstand sehen

- Bestellungen werden zunächst per mail und anruf gemacht

Daten ins Internet

- Betrieb darf nicht beeinträchtigt werden

- Level 0: Transfer per Extrakt + LOAD
  - „copy from … to file“, „copy from file to server im Web“
  - Komplette Entkopplung
  - Minimale technische Anforderungen „Dateikopieren“
  - Belastung bei Extrakt
  - Aktualität

Per Streaming-Replikation

- Transaktionsaktuell
- Kleinstmögliche Belastung operatives System
- Webserver Read only
- One Way
- Ganzer Cluster im Internet
- Kein Schmerz mit Schema
- Gleiche Systemumgebung erforderlich (Linux/linux)

Transfer per Trigger-Replikation

- Slony / Londiste
- Teile der Datenbank möglich
- Schemaänderungen machen Schmerzen
- Spürbare Belastung Master-System
  - Kodierung
  - Schreiblast

2ndQuadrant
Professional PostgreSQL

Next Generation Replikation = pg_logical

- WAL-Stream erweitert um einige Informationen
  - wal_level = logical
- Unterhalb von Logical in WAL files binäre Änderungen „in Datei 1804 schreibe an Position xFAB708E2 die Bytes AE BF 78 09…
- Logical decoding macht aus angereichertem WAL-Stream update / insert artige Befehle
- KEIN Query reengeneering!!
- Kein Reparsing erforderlich

it‘s logical, Captain

- Unterschiedliche PG Versionen möglich, solange >= 9.4
- Theorethisch unterschiedliche OS möglich
- Teile von Datenbanken möglich
- Geringe Mehrbelastung des Masternodes
- Schemaänderungen herausfordernd

it‘s not Query Duplication

- Volatile Funktionen werden eingefroren auf ihr Ergebnis
  - Insert into wuerfel (augenzahl) values (random())

UPDATE HANDTUECHER
SET PRICE=PRICE *1.05

→ numrec(handtuecher) changesets
→ Breite des Changesets = rowwidth(handtuecher)

Vertriebsbüro auf Mallorca

- HVVW 3.0 MM-Edition
- Lokales Lager
- Betreut Hotels auf der Insel
- Primär mit Handtüchern aus Vor-Ort-Lager
- Konsolidierte Umsatzreports gesucht
- Artikelstammdatenpflege, Marketing… Funktionen der Zentrale

2ndQuadrant
Professional PostgreSQL

Technik: PostgreSQL BDR

- BDR verfügbar als spoon zu 9.4
- Produktiver, 2ndQuadrant supporteter Einsatz
- Multi Master Replikation

  Rahmenbedigungen passen: HH betreut Kunden in D, Mallorca Kunde auf Mallorca.
- → Kaum Schreibüberschneidungen

Mehr über BDR

- Basiert auf logical decoding (PostgreSQL 9.4)
- Extrahiert wie pg_logical logische Changesets aus dem WAL
- Erweitert um Handling von Konflikten
- Erweitert um Globale Sequencen

No free lunch

- HVVW 3.0, MM Edition muss umfangreich angepasst werden
- „your holding it wrong“ - es gibt einige Aktivitäten, die in einer MultiMaster Umgebung schädlich sind
- Hier nur wenige Gedankenanregungen

Gedankenanregungen

- Eingrenzungen von Massenänderungen siehe pg_logical
- „reserviere 10 Handtücher in HH, halte reserviert, reserviere 5 Handtücher in M, rolle back oder commit“
- 2 Updates können den gleichen PK betreffen, 2 Inserts sollten dies nicht tun.

  → Updates eher konfliktwahrscheinlich

  - „Platz nehmen an frischem Gedeck ↔ gebrauchten Gedeck“