True Production Databases for Impatient Developers

Suny Kim, Jean-Charles Thomas
Autoscout24
München

Schlüsselworte:
Refresh von Entwicklungsumgebungen, Betankung mit Livedaten, Anonymisierung, RMAN duplicate, Data Pump, Endeca Search Index, Continuous Integration

Einleitung
Wir Datenbankadministratoren haben die Aufgabe, den Entwicklern und Testern livenahe Datensysteme bereitzustellen. Das war schon bei traditionellen Entwicklungszyklen eine Herausforderung. Autoscout24 setzt seit 2010 auf agile Entwicklung mit Continuous Integration und Release on Demand. Das verlangt eine noch schnellere Betankung der Entwicklungsumgebungen, möglichst ohne Unterbrechung.

Das Projekt „Livebetankung“
Es geht um die Übertragung der Livedaten auf Entwicklungsumgebungen. Das betrifft:

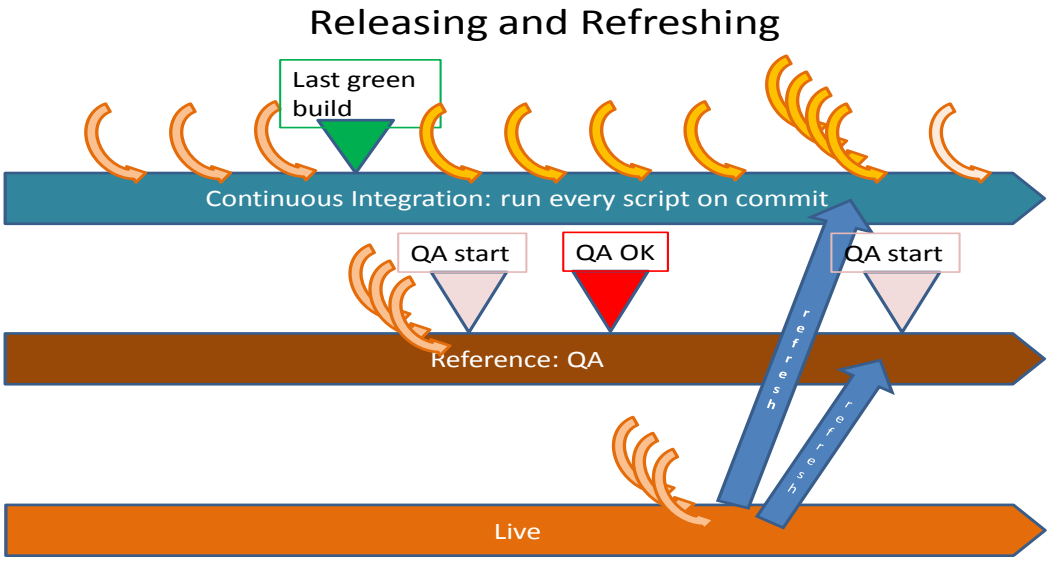
1. Die Datenstruktur (Tabellendefinition, Datenbankjobs u.ä.)
2. Die Daten: Wir reduzieren die Datenmenge nicht. Wir anonymisieren personenbezogenen Daten

Der Applikationscode ist nicht unsere Aufgabe.

Das Zusammenspiel mit den Releases
Wir haben vier Umgebungen:

1. Continuous Integration (CI), für die Entwicklung. Jedes neue Datenbankskript läuft, sobald es comitted ist.
2. Stable. Genauso wie CI.
3. Reference, für die finale Abnahme. Manueller Release. Häufigkeit: Zweimal täglich bis alle zwei Wochen.
4. Live. Manueller Release. Bis zu einmal täglich.

Wenn ein Scriptstand auf Live ist, ist er für CI und Stable bereits veraltet. Wir sind so gesehen immer zu langsam mit der Betankung. Um diese Lücke zu füllen, laufen nach der Betankung automatisch alle Skripte, die in der Zwischenzeit comitted wurden. Erst dann wird die Umgebung freigegeben.
Betankung und Release gehören also zusammen. Sie werden aus demselben Tool gesteuert (Team City).

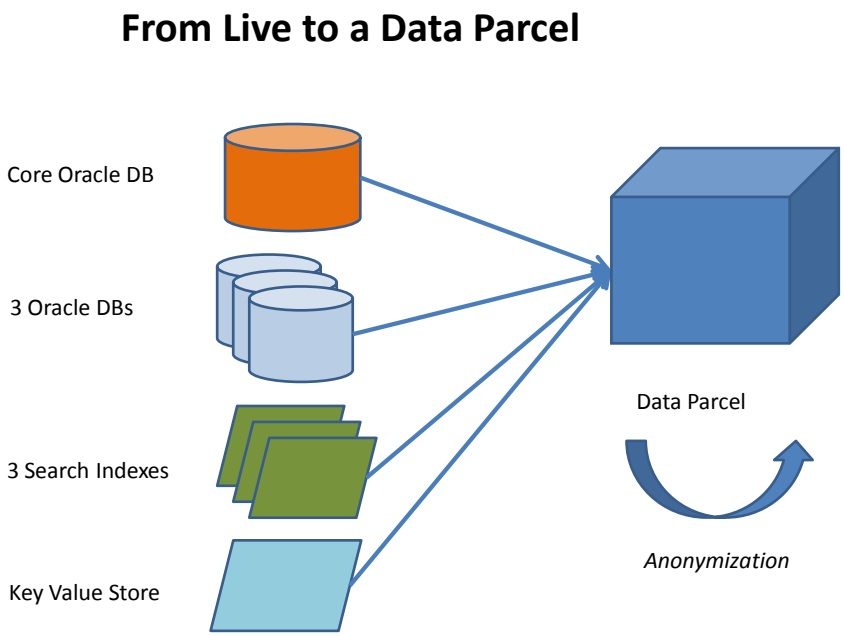


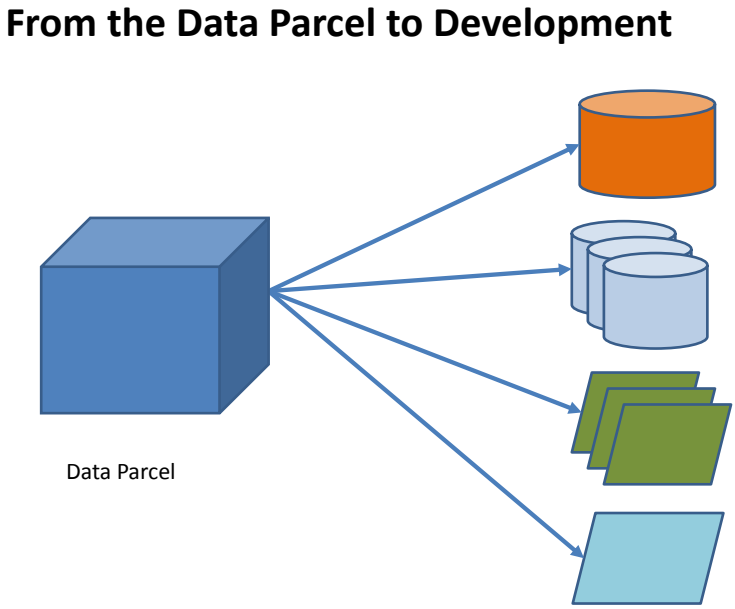
Die Schaukel-Architektur
Wir wollen die Unterbrechung durch die Livebetankung so kurz wie möglich halten. Deshalb haben wir alle Umgebungen doppelt aufgebaut: CI1 und CI2, Stable1 und Stable2, Ref1 und Ref2. Ein Loadbalancer führt die Applikation auf eine Umgebung, die damit aktiv ist. Sobald eine Umgebung passiv wird, wird sie im Hintergrund mit neuen Daten betankt und steht nach ein bis zwei Stunden bereit. Jetzt kann über den Loadbalancer auf die frisch betankte Umgebung switchen. Diese Architektur hat sich auch bei Wartungsarbeiten bewährt.

Die Data Stores
Unsere Daten leben in verschiedenen Systemen.

1. Die Kerndatenbank: Eine Oracle Datenbank mit Lesefarm (vergl. Vortrag „Active-DataGuard bei Autoscout24: eine Lesefarm im Praxiseinsatz“) (ca. 100GB)
2. Drei Randsysteme (Emails, Statistiken etc.): Oracle-Datenbanken (ca. 100GB)
3. Drei Search Indexes (ca.30 GB)
4. Ein Key-Value-Store (ca. 5 GB)

Wie können wir daraus ein Datenpaket schnüren, das in sich konsistent ist? Jede Betankung erzeugt einen Zeitstempel. Dann startet eine Kopie der abhängigen Systeme (Endeca, Key-Value-Store). Es folgen die Oracle-Datenbanken. Auf einem NFS-Server wird ein Verzeichnis mit dem Zeitstempel angelegt, darunter liegen Offline-Files zu jedem Data Store: Datenfiles der Kerndatenbank, Data Pump Exports der drei anderen Oracle-Datenbanken, Endeca-Indexes und ein Dump des Key-Value-Stores.
Wenn eine Entwicklungsumgebung betankt wird, werden die Dateien auf die Systeme gebracht. Die abhängigen Systeme (Standby, Endeca, Key-Value-Store) ziehen die Änderungen von der Kerndatenbank nach.





Die Steuerung

Es gibt

1. regelmäßige „Zwangsbetankungen“ nachts und am Wochenende

2. „Wunschbetankungen“ bei Bedarf, beispielsweise Tests auf der Referenz-Umgebung, angestoßen von uns Datenbankadministratoren oder den Release Engineers über ein Webtool. Oder zum wiederholten Testen von Datenmigrationen.

Die Techniken

Benutzte Oracletechniken sind:

1. RMAN duplicate – wird für die Kerndatenbank eingesetzt
2. Data Pump – wird für die anderen Oracle-Datenbanken eingesetzt
3. Selbstgeschriebene Prozeduren für die Anonymisierung

Aufgerufen und gesteuert wird das durch etwa 70 Shell- und SQL-Scripts mit über 10.000 Zeilen Code.
Warum so viel? Das Hauptskript für die Betankung der CI beispielsweise hat folgende Schritte:

- Inaktive Umgebung finden
- Monitoring stoppen
- Endeca-Index auf die Zielumgebung kopieren
- Master und Standby der Kerndatenbank erzeugen
- Die drei anderen Oracle-Datenbanken mit Data Pump Import füllen
- Die Endeca-Indexe nachziehen
- Monitoring starten

Die Anonymisierung braucht knapp 2000 Zeilen SQL-Code. Sie macht uns die größten Performance-Sorgen.
Wir haben uns gegen folgende Techniken entschieden:

4. SAN Snapshot
5. Flashback / Archive Apply

Performance

Alle Betankungsschritte sind voll automatisiert. Um die Betankung so kurz wie möglich zu halten, werden die einzelnen Datastores parallel betankt.
Die Referenz Umgebung wird mit den Originaldaten der Live Umgebung betankt, ohne Anonymisierung. Eine vollständige Betankung dauert etwa eine Stunde.
Für die CI und Stable Umgebungen werden erst die Daten von Live auf einem zentralen Server gespeichert und anonymisiert. Danach werden die einzelnen Umgebungen betankt. Eine vollständige Betankung einer Umgebung dauert etwa drei Stunden.

Erfahrungen, Erfolge und Niederlagen

- Die Betankung läuft stabil seit etwa einem halben Jahr
- Wir haben keine Probleme mehr durch unterschiedliche Stände der Umgebungen
- Die Belastung der Produktion durch RMAN duplicate und Data Pump ist nicht spürbar
- Kaum Unterbrechung durch die Betankung.

Aber:

- Ein Refresh im falschen Moment kann es sehr stören, beispielsweise während einer QA-Abnahme auf der Referenzumgebung. Deshalb wurde ein Reservierungssystem per REST-Interface eingerichtet.

- Eines der Randsysteme wächst so stark, dass wir das neue Techniken einführen müssen
- Die Hoffnung, das Ganze einfach zu halten, hat sich nicht erfüllt

Weiterentwicklung
Wir arbeiten ständig an Anpassungen und Erweiterungen und an Randapplikationen für bessere Bedienbarkeit:

- Wir entwickeln eine Front End Applikation für die Visualisierung der Umgebungen und deren Zustand (welche ist aktiv, welchen Daten-und Skriptstand haben die Systeme)
- Und eine zentrale Datenbank für die Steuerung der Betankung

Kontaktadresse:
Suny Kim
Autoscout24
Dingolfingerstr. 1-15
D-81673 München
Telefon: +49 (0) 89 444 56-1507
skim@autoscout24.com