

GoldenGate – Wie startet man so ein Projekt?

Mathias Zarick
Trivadis Delphi GmbH
Wien

Schlüsselworte

Replikation, Oracle, GoldenGate, Streams

Einleitung

Replikation ist nie leicht. Ein erstes Setup lässt sich nach absehbaren Aufwänden bewerkstelligen, eine Replikation am Leben zu halten ist jedoch die große Herausforderung. In diesem Vortrag, welcher auf einer großen Menge an Erfahrung mit Replikationsprojekten basiert, wird beleuchtet, welches die wichtigsten Schritte am Start eines Replikationsprojektes sind. Es werden auch essentielle Komponenten eines Replikationssetups beschrieben, um den Betrieb zu gewährleisten. Die Setups werden anhand von Oracle GoldenGate mit Oracle Databases erläutert.

Was ist Replikation und GoldenGate?

Laut dem englischen Wikipedia Eintrag für Replikation ist sie durch folgende Kriterien charakterisiert: „sharing information so as to ensure consistency between redundant resources“ und „improve reliability, fault-tolerance, or accessibility“. Es geht hier also um die Zurverfügungstellung von Informationen an verschiedenen Ort/Quellen etc. also für uns Datenbanken, wobei ein ganz wichtiger Punkt ist, auf Konsistenz nicht nur zu achten, sondern sie auch sicherzustellen. Ebenso lassen sich Verlässlichkeit, Ausfallsicherheit und Zugriffsmöglichkeiten steigern, kurzum die Verfügbarkeit kann oder soll sogar gesteigert werden.

Mit der Oracle Datenbank lassen sich logische und physische Replikationen bewerkstelligen. Logische sind solche, die in eine Replika DB SQL einspielen, welches auf einer Datenquelle gesammelt (oder gecaptured) wurde. Physische Replika sind solche, die durch Block Recovery 1:1 Abbilder auf Stand halten. Dazu gehört die physical standby Datenbank. Physical Standby Datenbanken sind im Betrieb „pflegeleichter“ als logische, da für solche mehr Restriktionen existieren und andere Anforderungen an die Ressourcennutzung und Effizienz ebendieser gestellt werden. Es gibt noch weitere Gründe, warum es schwieriger ist, eine logische Replikation zu betreiben, welche im Laufe des Vortrages beleuchtet werden.

Für Oracle Datenbanken begann die logische Replikation mit Snapshot Replikation, später Materialized View Replikation, später gab es dann die Advanced Replication, welche schon sehr anspruchsvolle Setups ermöglichte. Logical Standby wurde mit 9iR2 eingeführt und erlaubte es Standby Datenbanken nicht nur im mount Status, sondern geöffnet zu betreiben, so konnten dort Queries ausgeführt werden. Oracle Streams wurde ebenso mit 9i eingeführt, zuverlässig funktionierte es eigentlich erst mit 10.2.0.4. Es basiert ebenso wie Logical Standby auf dem LogMiner. Dieser extrahiert SQL, welches dann u.U. transportiert oder propagiert und wieder eingespielt werden kann. Oracle GoldenGate wurde seitens Oracle durch den Ankauf der Firma GoldenGate in das Replikations-Portfolio integriert und stellt derzeit die strategische Lösung auf diesem Gebiet dar. Seitens Drittherstellern möchte ich Shareplex von Dell und DataMirror nennen. Letzteres wurde von IBM gekauft und später wieder an die Firma Rocket Software veräußert.

Replikationen können helfen, um Zero-Downtime-Migrationen zu vollziehen, z.B. Änderung des OS, des Character Sets, der DB Version, der Endianness der Datafiles, etc.. Ebenso können Replikationen als Hochverfügbarkeitslösung eingesetzt werden, Replikationen arbeiten sinnvollerweise asynchron, und erlauben so die unabhängige Arbeit an mehreren Replikas, selbst wenn die Verbindung dazwischen unterbrochen ist. Man kann Replikationen ebenso dazu verwenden, um produktive OLTP Systeme zu entlasten (offload). Dieser Fall lässt sich allerdings ebenso mit einer physikalischen

Replikation mit Active Data Guard bewerkstelligen, was daher bevorzugt werden sollte. Andere Anwendungsfälle sind das Historisieren oder auf das Zentralisieren von Daten.

Warum sollte man nun auf GoldenGate setzen. Als Pro sind folgende Fakten zu nennen: Es ist die aktuelle und am meisten gepflegte Replikationstechnologie von Oracle. Es hat bessere Performance im Vergleich zu Streams. Vorsicht: das gilt nur für die klassischen Prozesse! Es bietet heterogene Replikation, beispielsweise von DB2 nach Oracle. Es werden im Vergleich zu Streams nur solche Transaktionen übermittelt, welche auch committed wurden, was man genaugenommen eigentlich auch als Nachteil auslegen kann, denn so gehen die Transaktionen erst komplett auf den Weg, wenn das Commit erkannt wurde, kommen also später an. Als Nachteile vor allem gegenüber Streams sind folgende Punkte zu nennen: Es ist separat zu lizenzieren. Es gibt ein eigenes proprietäres Konfigurationinterface. GoldenGate Setups können sehr schnell sehr komplex werden. Es müssen externe Prozesse betrieben werden.

Die GoldenGate Architektur sieht ähnlich der von Streams aus. Es gibt Prozesse für das Sammeln von Daten (Capture oder Extract), für das Weiterleiten von Daten (Propagation oder Data Pump Extract) und für das Einspielen von Daten (Apply oder Replicat). GoldenGate produziert dabei sogenannte Trails, die im Filesystem gespeichert werden, bevor sie transportiert und/oder konsumiert werden. Die Prozesse Extract bzw. Replicat lassen sich als klassische Prozesse oder als integrierte Prozesse betreiben. Letzteres ist Stand heute immer zu empfehlen, da beim integrierten Betrieb die wenigsten Einschränkungen existieren und eine Integration in HA Setups wie RAC oder Data Guard auch kein Problem darstellen. Leider ist dadurch auch die gute Performance, die die klassischen Prozesse hatten, gebremst, denn wir kommen damit wieder zu einer Produktion von LCRs mit dem LogMiner wie bei Streams. Auf der Zielseite wird ein DB interner Apply wie bei Streams betrieben.

Regeln für ein erfolgreiches Replikationssetup

Folgendermaßen sollte im Projekt vorgegangen werden:

1. Ziel des Vorhabens muss klar definiert sein: „Warum braucht man die Replikation?“.
2. Es sollte ein „Proof of Concept“ durchgeführt werden.
3. Die Topologie und das Setup müssen passend zum Scope entworfen werden.
4. Es sollen Regeln und Prozesse für die Replikationsumgebung entworfen werden, z.B. wie ist mit DDL umzugehen, wie soll das Replikationssetup bei Bedarf erweitert werden.
5. Es muss ein gut getestetes Set an Scripts, Programmen, Dokumentation erstellt werden, mit welchem man die Replikation (teil)automatisiert aufsetzen und betreiben kann.
6. Wenn alle vorherigen Punkte ausreichend erfüllt sind darf man in den Betrieb gehen.

Während des Betriebs müssen dann ständig Schwächen im Setup und Monitoring erkannt und verbessert werden. Dafür sind passende Testsysteme unerlässlich.

Folgende 8 Regeln sollten im Projekt von Anfang an beachtet werden:

1. KISS – Keep it Small and Simple! Bzw. Keep it Simple, Stupid! Achten Sie darauf, dass sie es nur so komplex machen, wie es wirklich nötig ist.
2. Datenbanken müssen korrekt konfiguriert werden. Force Logging und Archivelog mode sind auf der Quelle neben minimal supplemental logging unerlässlich. Hingegen sollten sie zusätzliches supplemental logging nur so granular aufsetzen, wie es die Replikation wirklich braucht, um unnötige Overheads zu vermeiden.
3. Kritische Komponenten sind unerlässlich. Dazu gehören die letzten empfohlenen GoldenGate Patches, ein Heartbeat Monitor und verschiedene weitere Monitore (Prozesse, Log Files, Ressourcennutzung, Transport Lag, Apply Lag). Ebenso wird empfohlen, ein eigenes Repository zu betreiben, welches das Replikationssetup abbildet und von welchem die Setupscripte generiert werden können.

4. Man sollte in der Lage sein, die replizierten Daten zu verifizieren, d.h. zu überprüfen, ob die Konsistenz wirklich gewährleistet ist. D.h. man muss die Daten vergleichen, gegebenenfalls auch ausgleichen können.
5. Man sollte in der Lage sein, ein Re-Setup der Replikation anzustoßen und die Daten zu synchronisieren.
6. Man sollte verschiedene Recoveries der Datenbanken (komplett, point-in-time) und deren Einfluss auf das Setup testen. Anschließend muss das Replikations-Setup wiederhergestellt werden. Das Ganze ist gut zu dokumentieren.
7. Training für den Betrieb ist das A und O. Eine Installation und Deinstallation sollte mehrfach getestet werden, bevor man produktiv geht. Ein Testsystem soll verlässlich aus dem Produktionssystem geklont werden können und dabei ein ebenso repräsentatives Replikationssetup darstellen. Mit Hilfe des Testsystems lassen sich dann auch Fehlerszenarien wie zum Beispiel Konflikte nachstellen.
8. Die Dokumentation des Setups und Betriebs ist sehr wichtig. Einzelne Handgriffe, um Maßnahmen im Betrieb durchzuführen sollten genauestens beschrieben sein.

Konfliktauflösung

Ein Konflikt entsteht, wenn einer oder mehrere Datensätze an mehreren Orten bzw. Replikas (nahezu) gleichzeitig verändert werden. Da es kein globales Locking und eine asynchrone Übertragung gibt, kann es zu solchen Phänomenen kommen. Konflikte gefährden die Konsistenz, sollen daher vermieden, mindestens aber erkannt und dann auch gelöst werden.

Konflikte sollten am besten durch ein gutes Applikationsdesign vermieden werden. Zum Beispiel, dahingehend, dass gewisse Daten nur an einem gewissen Ort verändert werden dürfen. Partitioniert man dahingehend seine Daten ausreichend, ließen sich Konflikte schon dadurch vermeiden.

Eindeutigkeitskonflikte sind einfach zu vermeiden, es gibt genügend Mechanismen, um eindeutige IDs für neue Datensätze zu generieren, z.B. die SYS_GUID Funktion oder ein einfaches Inkrement bei Sequenzen.

Wenn Konflikte nicht ausgeschlossen werden können, so muss man sie zumindest erkennen.

GoldenGate hat hier einen entscheidenden Nachteil im Vergleich zu Streams und Advanced Replication. Denn in den Default Setups werden lediglich Primary Key Werte verglichen, um Konflikte zu erkennen, d.h. viele Konflikte werden gar nicht erst erkannt. Die Konsistenz steht dann auf dem Spiel. Man muss auf eine korrekte Parametrisierung achten, um eine akkurate Konflikterkennung zu implementieren, bei welcher nicht nur Primary Key Werte sondern alle geänderten Spaltenwerte miteinander verglichen werden.

Mit Streams war man es gewohnt, dass es eine zentrale Stelle mit Replikationsfehlern gab, welche man einfach per SQL abfragen kann (dba_apply_error). So etwas gibt es bei GoldenGate nicht von Haus aus. Man kann es lediglich relativ mühsam über sogenannte Exception Tables nachbauen, wobei man für volle Nachvollziehbarkeit eine solche Table sinnvollerweise für jede replizierte Table bräuchte, was schnell sehr aufwändig werden kann.

Wie sind die Empfehlungen rund um automatische Konfliktauflösungen:

In jedem Falle sollte eine exakte Konflikterkennung konfiguriert werden, d.h. es müssen auch die Before-Images der geänderten Spalten verglichen werden. Wenn es keine Multimasterreplikation gibt, so sollte man auch keine Exception Tables oder Auflösungen konfigurieren, im Falle eines erkannten Fehlers sollte der Replicat abgebrochen werden.

Wenn es eine Multimasterreplikation und damit ein Potenzial für Konflikte gibt, so sollte man zunächst einmal ohne Konfliktauflösungen beginnen. Wenn es nun zu Konflikten kommt, so sollte man jeden einzelnen verstehen und mit Hilfe eines angepassten Applikationsdesign vermeiden, insofern das möglich ist. Betroffene Tables sollten dann exklusive Exception Tables bekommen. Zunächst sollten dann solche Konflikte, die wirklich nicht durch die Applikation vermieden werden können, manuell aufgelöst werden. Zu guter Letzt wird man doch noch ein paar automatische Konfliktauflösungen implementieren. Aber nur für folgende Fälle: Konflikte, für welche man weiß,

warum sie passieren, welche man nicht applikatorisch vermeiden kann und für welche man sicher weiß, wie sie richtig aufzulösen sind. Denn eines muss man sich immer vor Augen halten. Eine automatische Konfliktauflösung kann zu Konvergenz und damit Konsistenz führen, aber es gibt immer einen Verlierer. Gewisse applikatorische Änderungen mussten abgelehnt, überschrieben nochmals angepasst etc. werden.

Einige weitere Tipps

Für die Oracle Datenbank ist man es gewohnt, die offizielle Dokumentation lokal auf seinem Laptop für das Offline Browsing abzuspeichern. Dafür lädt man entsprechende zip Archive vom OTN herunter. Solche Archive sucht man für GoldenGate vergeblich, zumindest für die aktuellen Versionen. Mit folgendem Befehl kann man sie jedoch lokal ablegen, vorausgesetzt man hat das Programm wget zur Verfügung.

```
wget \  
  --recursive \  
  --no-clobber \  
  --page-requisites \  
  --html-extension \  
  --convert-links \  
  --restrict-file-names=windows \  
  --domains docs.oracle.com \  
  --no-parent \  
  http://docs.oracle.com/goldengate/1212/gg-winux/index.html
```

Das Tool ggsci, welches für die Administration von GoldenGate verwendet wird, verlangt komischerweise, dass man in dem Pfad steht, wo es selbst abgelegt ist. Mit folgendem Wrapper Shell Script lässt sich das ganze „Problem“ lösen, gleichzeitig wird auch noch rlwrap für die Historie von Befehlen integriert.

```
#!/bin/sh  
oldpwd=$(pwd)  
cd ${GG_HOME}  
rlwrap -i -f ${GG_HOME}/ggsci.key ./ggsci  
cd ${oldpwd}
```

Fazit

GoldenGate ist eine fortschrittliche Replikationstechnologie. Einige Dinge, die mit Streams bzw. Advanced Replication einfach so funktionierten erscheinen mit GoldenGate jedoch etwas kompliziert, zum Beispiel Konflikterkennung und -auflösung. Die in dem Vortrag erwähnten 8 Regeln sollten beherrzt werden, wenn man „Spaß“ an seinem Setup haben möchte.

Kontaktadresse:

Mathias Zarick
Trivadis Delphi GmbH
Millennium Tower
Handelskai 94-96
A-1200 Wien

Telefon: +43 1 332 35 31 46
Fax: +43 1 332 35 34
E-Mail: mathias.zarick@trivadis.com
Internet: www.trivadis.com