

GoldenGate und Siebel in Zero-Downtime

Shane Gilbride, Sebastian Knoblich und Stefan Wurm, Cirquent GmbH

Oracle bietet eine Vielzahl an Produkten zum Verschieben und Replizieren von Daten an. Nicht alle genügen den Anforderungen von Zero-Downtime in einer Hochverfügbarkeitsumgebung. Dieser Artikel betrachtet Oracle GoldenGate, eine Hochleistungs-Software zur Echtzeit-Replikation von Daten, in einer hochverfügbaren Siebel-Umgebung.

Die Aufgabe scheint zuerst einfach: Implementierung einer Siebel-CRM-Anwendung in einer hochverfügbaren und hochintegrierten System-Umgebung. Die Migration der riesigen Datenmengen wäre hierbei eigentlich immer die Angelegenheit des Enterprise Integration Managers (EIM). Das geht jedoch nicht mit der Anforderung, das System unter allen Umständen (Deployment, Migration, Upgrade) 24/7/365 zu betreiben. Hierbei stößt man an die Grenzen von EIM. Die Frage ist nun, welche weiteren Möglichkeiten Oracle derzeit anbietet, um diese Anforderung zu erfüllen (siehe Abbildung 1).

Der Oracle Active Data Guard zum Kopieren von Datenbanken mit geringer Latenzzeit, jedoch ohne Transformationsmöglichkeiten, ist als Datensicherheits-Tool lediglich zur Minimierung des Ausfallrisikos hilfreich. Der Oracle Data Integrator (ODI), eine ETL-Software (Extract, Transform and

Load) als zentraler Bestandteil von Oracle Fusion, stellt sehr gute Transformationsmöglichkeiten zur Verfügung, bietet aber wiederum nicht die notwendige Synchronisation in Echtzeit.

Die durch Oracles Akquisition im Jahr 2009 hinzugekommene GoldenGate-Software, einem in Echtzeit arbeitenden Hochleistungsreplikations- und Transformations-Tool, sollte zur Erreichung der Zero-Downtime beitragen. GoldenGate arbeitet auf der Datenbank-Transaktionsebene und erlaubt dort sowohl den Austausch als auch die Transformation von Daten zwischen Datenbanken. Die modulare Architektur bietet die nötige Flexibilität, die zur Extraktion und Replikation ausgewählter Datensätze nötig ist. Dies geschieht mittels der Erfassung von transaktionalen Modifikationen durch Data Manipulation Language (DML) und von strukturalen Änderungen mittels Data Definition Language

(DDL) über heterogene System- und Anwendungs-Umgebungen hinweg.

Dieser Artikel beschreibt die Grundlagen von GoldenGate und im Speziellen dessen Verwendung für Zero-Downtime-Migrationen. Dabei wird zum einen ein Konzept vorgestellt, das die Migration von hoch integrierten und komplexen Systemen erlaubt. Zum anderen werden sowohl die Herausforderungen und Probleme in Bezug auf eine Siebel-Migration mit GoldenGate als auch deren Lösung beschrieben.

Die GoldenGate-Architektur

GoldenGate stellt ein flexibles und mächtiges Framework/Pattern zur automatischen Replikation von Daten-Änderungen zur Verfügung. Ein dreigliedriger Prozess bestehend aus „Extract“, „Pump“ und „Replication“ (siehe Abbildung 2) macht es möglich, Änderungen verschiedenster Art flexibel aufzugreifen und zu verteilen:

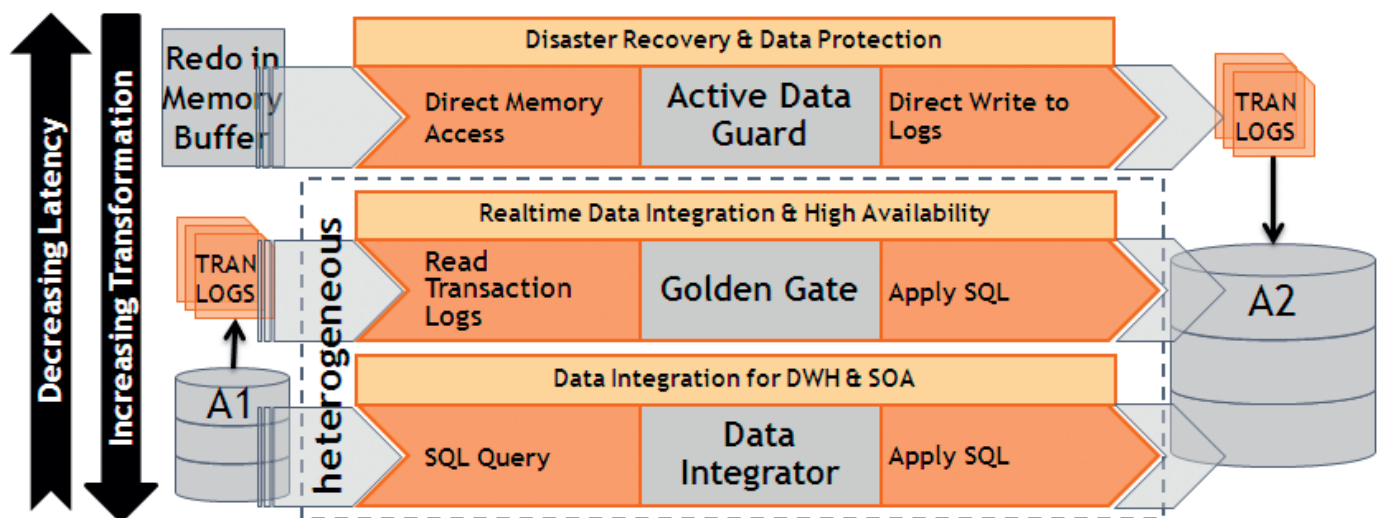


Abbildung 1: Oracle-Produkte zur Datenreplikation

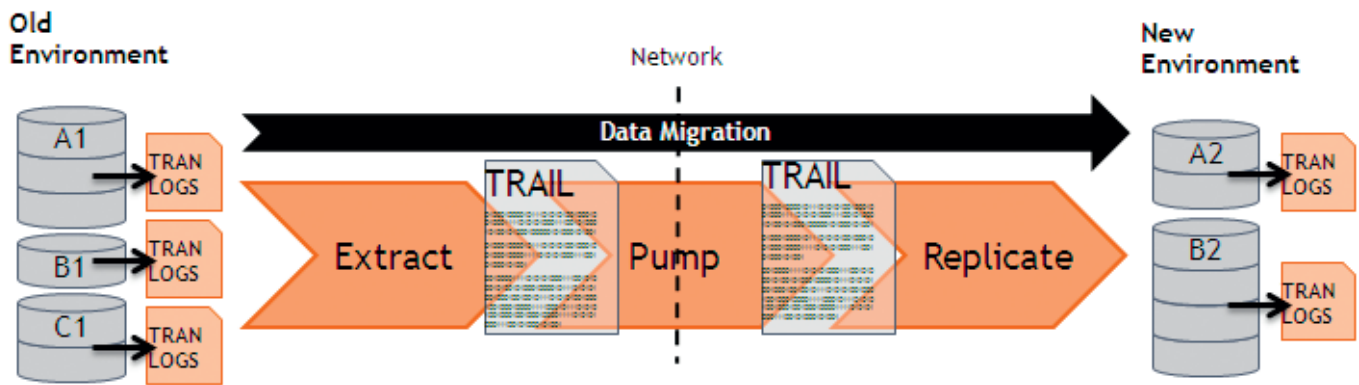


Abbildung 2: GoldenGate-Architektur und -Prozesse

- Transaktionale Datenänderungen (DML)
- Ausgewählte Daten (mittels Filterung von Tabellen und Tabellenspalten)
- Änderungen des Datenbank-Schemas (DDL)

Ein Extract-Prozess liest von den sogenannten „Transaction Logs“ (für die Delta-synchronisation) oder direkt aus der Datenbank (für Initialsynchronisation) und schreibt die Daten in eine lokale Trail-Datei. Listing 1 zeigt einen einfachen Extract-Prozess mit dem Namen „ext01“, der alle Änderungen für bestimmte Tabellen erfasst.

```
EXTRACT ext01
USERID ggs, PASSWORD doag
EXTTRAIL /trails/lc
TABLE siebel.S_CONTACT;
```

Listing 1

Bevor der Extract gestartet wird, ist es notwendig, die zusätzliche Protokollierung (Supplemental Logging) der Datenbank zu aktivieren und den eindeutigen Schlüssel jeder Tabelle für jeden Datensatz in das Transaction-Log zu schreiben. Ein Pump verteilt die Trail-Dateien über das Netzwerk an die Zielsysteme. Listing 2 zeigt, wie man alle Daten im Schema „Siebel“ aus der Trail-Datei an das jeweilige Zielsystem versendet.

```
EXTRACT pump01
PASSTHRU
RMTHOST target.host.de, MGRPORT
7809
RMTTRAIL /trails/rt
TABLE siebel.*;
```

Listing 2

Ein Pump-Prozess ist im Wesentlichen ein Extract-Prozess mit dem speziellen Parameter „PASSTHRU“, der den Extract in den Pump-Modus setzt. Schlussendlich werden die Daten transformiert und auf die Zieldatenbank angewendet. Listing 3 ist ein Beispiel dafür. Die „USEDEFAULT“-Option des „COLMAP“-Parameters definiert, dass Tabellenspalten mit gleichem Namen und Format automatisch zwischen Quell- und Ziel-Tabelle einander zugeordnet werden.

Im Allgemeinen gibt es keine Einschränkung in GoldenGate, wo die Transformation durchgeführt wird. Diese kann entweder wie zuvor beschrieben auf dem Zielsystem, aber auch auf dem Quellsystem oder auf einem intermediären System durchgeführt werden. Die Entscheidung ist abhängig von der Arbeitslast und den Anforderungen hinsichtlich der Leistung/Performance der involvierten Systeme.

Zwischen den Prozessen „Extract“, „Pump“ und „Replicate“ befinden sich Trail-Dateien (siehe Abbildung 2). Diese sind eine Sicherung für die Prozessierung der Daten im Falle von Unterbrechungen der Verbindung oder des Ausfalls des Prozesses auf dem Zielsystem. Eine Trail-Datei ist eine flache/einfache (Text-)Datei, die GoldenGate-Daten in einem generischen Format enthält. Die Verwendung von Trail-Dateien führt zu einer lose gekoppelten und robusten Architektur. Es spielt dabei keine Rolle, ob es sich um einen Fehler im Pump, im Netzwerk oder im Replicate-Prozess handelt. Der Prozess kann mithilfe der Trail-Dateien stets exakt von dem Punkt aus fortfahren, an dem der Prozess unterbrochen wurde. Zusätzlich garantiert die Aufbewahrung der Archive Logs (Log Retention) für den Extract-Prozess, dass selbst der Ausfall

einer Datenbank den Replikations-Prozess schadlos, vollständig und lückenlos aufrechterhält.

Migrationskonzept der reduzierten Funktionalität

Der Erfolg einer Strategie von Zero-Downtime-Deployments hängt – wie eingangs erwähnt – maßgeblich von der Komplexität der beteiligten Systeme und der System-Landschaft ab. Um nun in solch hoch integrierten und komplexen System-Umgebungen eine Zero-Downtime-Migration durchzuführen, bedarf es eines Konzepts, das die Migration in drei leichter durchführbare Phasen aufteilt, jede mit einer genauen und detaillierten Liste an durchzuführenden Aufgaben (siehe Abbildung 3):

- T-1: Zeitdauer vor dem Umschalten (Switch Over) auf die neue Umgebung, die für alle integrierten Systeme einen reduzierten Funktionsumfang definiert, mit dem die hauptsächlichen Geschäftsprozesse noch durchgeführt werden können. Wo immer möglich, sollten Prozesse in eine Warteschlange gesetzt und wiederkehrende, geplante Jobs verschoben werden.
- T-0: Zeitpunkt der Umschaltung (Switch Over)
- T+1: Zeitdauer, um alle Systeme auf volle Funktion zu schalten, anschließend Migrations-Prozesse, die nicht vorher durchgeführt werden konnten, nachzubearbeiten sowie Prozesse in der Warteschlange und geplante Jobs durchzuführen.

Idealerweise beträgt die Zeit für die Phasen „T-1“ und „T+1“ je nach Aufwand nur einige wenige Stunden. Dadurch dass GoldenGate

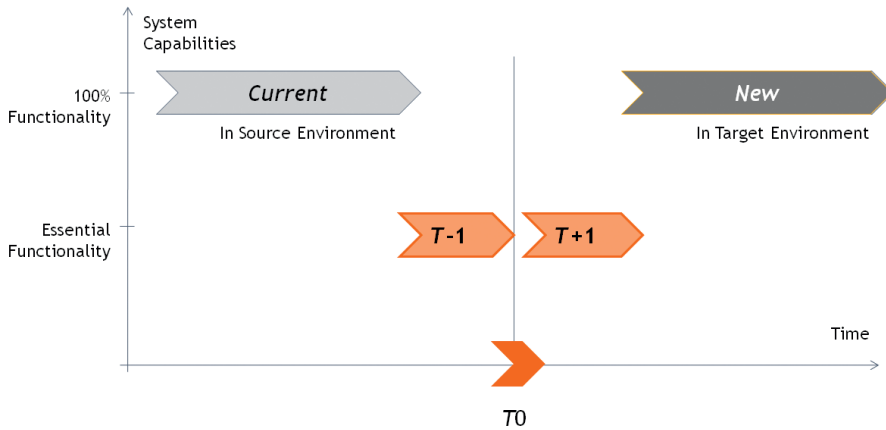


Abbildung 3: Konzept – Zero-Downtime-Migration

```

REPLICATE rep01
USERID ggs, PASSWORD doag
MAP siebel.S_CONTACT, TARGET siebel.S_CONTACT
COLMAP (
  USEDEFAULT,
  X_LONG_NAME = @STRCAT (LAST_NAME, FIRST_NAME),
  X_REGION = @IF ( @VALONEOF (STATE, „CA“, „AZ“, „NV“,
                              „WEST“, „EAST“)
);

```

Listing 3

```

MAP siebel.S_CONTACT, TARGET siebel.S_SRC_PAYMENT,
FILTER (ON INSERT),
SQLEXEC (
  ID next_id,
  QUERY="select siebel.s_sequence_pkg.get_next_rowid
        from dual",
  NOPARAMS
),
COLMAP (
  ROW_ID = @GETVAL (next_id.get_next_rowid),
  TYPE = "Cash On Delivery"
);

```

Listing 4

die neue Umgebung zu jedem Zeitpunkt in „T-1“ synchron hält, dauert die eigentliche Umschaltung (Switch Over) der Systeme nur wenige Minuten, in denen die Benutzer sich am alten System ab- und am neuen wieder anmelden.

Siebel-spezifische Probleme und Lösungen mit GoldenGate

Überprüfung von Transformation und Mapping: Jede Transformation und jedes Mapping wird zusammen mit Oracle Expert Services analysiert und verifiziert. Der Grund hierfür liegt in der eigentlichen Bestimmung von Siebel und dem Verbot der Manipulation von Daten direkt per SQL. Da

GoldenGate im Kern nichts weiter als direkt SQL gegen die Datenbank absetzt, ist eine solche Überprüfung nicht nur zu empfehlen, sondern auch notwendig.

Erstellung neuer Siebel-Row-Ids: Das Generieren neuer Datensätze aus GoldenGate heraus erfordert in Siebel auch die Generierung einer neuen Row-Id. Dies ist mithilfe des Siebel Sequence Package möglich. Es erlaubt mittels eines Datenbank-Skripts die Erzeugung einer eindeutigen Row-Id. Listing 4 veranschaulicht, wie eine neue Zahlungsmethode „Cash On Delivery“ für alle neuen Kontakte automatisch erstellt werden soll. Diese wird auf der Ziel-Umgebung für jeden neuen Kunden erzeugt.

Vermeidung von Duplicate-Key-Generierung: Synchronisiert man mit GoldenGate zwei Siebel-Anwendungen, werden auch alle Row-Ids der Quell-Umgebung identisch in die Ziel-Umgebung repliziert. Dies führt zu Problemen, wenn in der Ziel-Umgebung nach dem Go Live neue Datensätze angelegt werden. Um die Generierung identischer Row-Ids in diesem Fall zu vermeiden, kann man den Corporate-Präfix im Ziel-System ändern. Dies hat den weiteren Vorteil, dass man die Erstellung des Datensatzes über mehrere Migrationen hinweg nachverfolgen und dadurch feststellen kann, aus welchem Release der Datensatz ursprünglich kam.

Vermeidung von doppelter Datenverarbeitung: Doppelte Prozessierung von Daten im Zielsystem muss beachtet werden. Hierzu zählen unter anderem Workflow Policies, Assignment Policies und Siebel Repeating Batch Jobs, die im Detail analysiert und gegebenenfalls ausgeschaltet werden müssen.

Siebel Referenz- und Konfigurations-Daten: Referenz- und Konfigurations-Daten (wie List of Values, Organisations/Divisions, Positions, Responsibilities) sollten generell repliziert werden, damit die entsprechenden technischen Schlüssel erhalten bleiben. Ausnahmen bei der Replikation sind im Detail zu betrachten und hierbei ist dann oftmals ein komplexeres Mapping zwischen altem und neuem Siebel-System notwendig.

Fazit

Geplante Ausfallzeiten von Systemen im Zuge von Upgrades, Migrationen oder Systemwechsel (Cutover) sind jederzeit kritisch zu betrachten. Die Durchführung der Änderungen kostet nicht nur Zeit und Geld, sondern ist auch mit indirekten Kosten in Verbindungen zu bringen.

Unter Berücksichtigung der Besonderheiten des zu migrierenden Systems und gleichzeitiger Reduktion der Komplexität durch die zeitweise Aussetzung von nicht geschäftskritischer System-Funktionalität während des Systemwechsels sind Zero-Downtime-Deployments für Siebel CRM Systeme möglich.

Shane Gilbride
shane.gilbride@Cirquent.de