Extraktion und Laden im Oracle Data Warehouse – Varianten und Möglichkeiten

Peter Welker
Trivadis GmbH
Stuttgart

Schlüsselworte:

Oracle, Data Warehouse, ETL, Extraktion, Laden, Data Guard, Transportable Tablespace, Golden Gate, Journal, ORA_ROWSCN, Change Data Capture

Einleitung

Leider wird die Datenextraktion aus den Quellsystemen und das Laden dieser Daten in die Staging Area eines Data Warehouses oft ziemlich stiefmütterlich behandelt. Tägliche Komplettabzüge einzelner Tabellen per Database Link oder deren Export in eine CSV Datei sind die üblichen Methoden. Alternativ verwendet man einen speziellen Adapter bspw. für den Zugriff auf SAP – und nimmt auch hier alles mit, was das Quellobjekt so zu bieten hat. Warum auch nicht? Durch einen geeigneten Abgleich mit dem Core-DWH lassen sich so die relevanten Daten seit dem letzten Abzug ermitteln; je nach Datenmenge und Methode manchmal sogar in annehmbarer Zeit.

Schon etwas diffiziler wird es, wenn die Quelle einfach zu viele Daten liefert und so aus Zeit- oder Ressourcengründen der Abzug einer Teilmenge der Daten notwendig wird. Aber wie ermittle ich die seit der letzten Extraktion geänderten Daten ohne den Vergleich mit den Daten im Data Warehouse? Glücklicherweise hängt ja meist eine Zeitstempel-Spalte an den Quelldatentabellen. Wenn man sich den letzten Abzugszeitpunkt gemerkt hat, muss man nur die Datensätze selektieren bei denen der Wert dieser Spalte größer ist, oder? Im Prinzip ja, aber wie ermittle ich damit die gelöschten Daten? Und die mehrfachen Änderungen von Datensätzen innerhalb des Tages? Und wie stelle ich sicher, dass ich nichts vergessen habe? Der Zeitstempel eines Datensatzes kann ja um 23:59:55 Uhr geändert aber erst um 00:01:30 Uhr committet worden sein. Und wenn meine Extraktion immer pünktlich um 00:00:00 Uhr startet habe ich ein kleines Problem …

Kurzum, es ist an der Zeit eine Lanze für einen ernsthafteren Umgang mit der Datenextraktion und dem anschließenden Ladeprozess zu brechen. Und genau wie im Handwerk gibt es für jede Anforderung das geeignete Werkzeug und den richtigen Umgang damit – nur MacGyver löst alle Probleme mit Taschenmesser, Bindfaden und Kaugummi …

Werkzeuge und Methoden im Überblick

Wir betrachten dabei grob gesagt folgende Werkzeuge und Methoden:

- **Extraktion**
  - „Vollabzüge“, also die Extraktion aller Datensätze relevanter Entitäten im Quellsystem bei jeder Aktualisierung
    - „SELECT ohne Filter“, Standby DB oder Transportable Tablespace
  - Varianten, die geänderte Daten im Quellsystem identifizieren

        - Änderungsspalten, Journaltabellen oder ORA_ROWSCN
        - Programmierte Trigger, Data Guard, Oracle Change Data Capture oder Oracle Golden Gate
- **Transport & Laden**
    - File-Transfer inkl. SQL*Loader, External Tables und XML Ladeverfahren
    - Datenbank-Link
    - Transportable Tablespace
    - Logfile Transfer
    - ETL Werkzeuge
    - SOA Technologien
    - Messaging Systeme

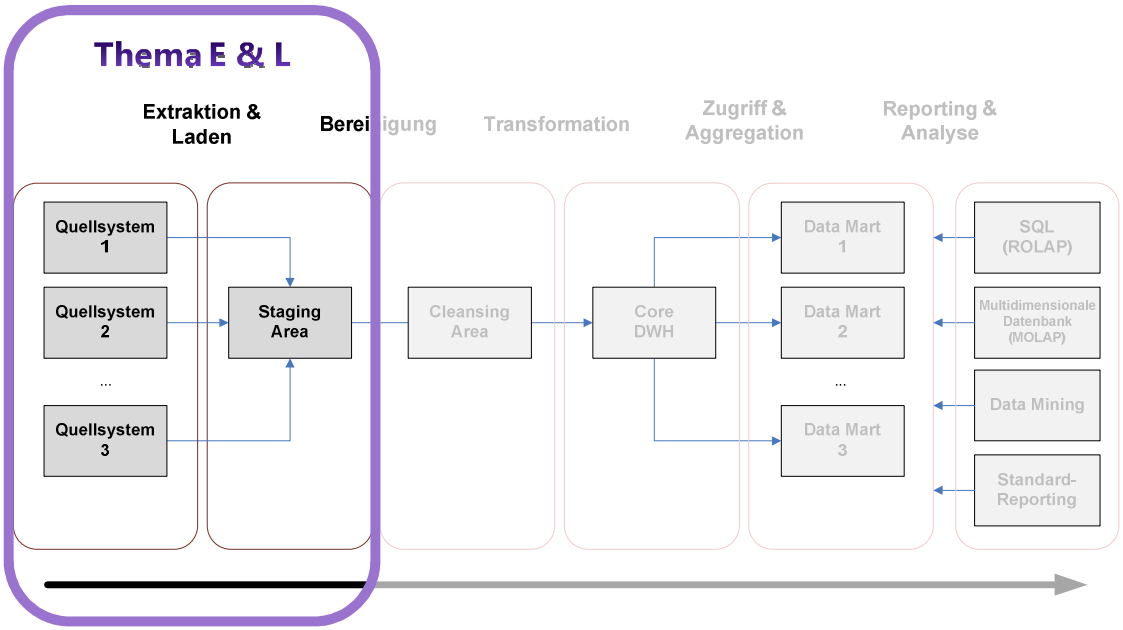

Abbildung 1: Extraktion und Laden im Umfeld der Data Warehouse Prozesse

Anforderungen

Identifikation, Extraktion, Transport und Laden. Das sind die Themen, die uns in diesem Vortrag beschäftigen. Beginnen wir mit der Extraktion der relevanten Daten.

Identifikation und Extraktion

Nehmen wir an, wir wollen zunächst die Daten der Kunden und Ihrer Bestellungen aus einem Quellsystem ins Data Warehouse übertragen. Da stellen sich gleich mal ein paar Fragen:

- **Frequenz**: Wie oft – und wann – werden Änderungen aus dem Quellsystem ins DWH übertragen?
- **Latenz**: Welche Zeit darf das DWH benötigen, um eine Änderung im Quellsystem für die Endanwender verfügbar zu machen?

- **Vollständigkeit der Historie**: Ist es ausreichend, einen „Schnappschuss“ zum Zeitpunkt der Extraktion mitzunehmen, oder sind mehr Informationen erforderlich? Hier gibt es mehrere Kriterien
    - **Änderungen:** Sind alle Änderungen zwischen zwei Extraktionszeitpunkten wichtig?
    - **Löschungen**: Ist ein Schnappschuss ausreichend sofern er auch Informationen über gelöschte Daten beinhaltet?

Die Antworten auf diese Fragen führen direkt zu den passenden Extraktionsmethoden. Vielleicht muss dafür das Quellsystem angepasst, eine weitere Datenbankfunktionalität genutzt oder sogar eine neue Software angeschafft werden um via LogFile-Analyse oder gar per Trigger die benötigten Datenänderungen zu identifizieren?

Bei Methoden, die Dateninkremente erzeugen – also Änderungen identifizieren können – ist natürlich auch eine klare Abgrenzung der einzelnen Extraktionsvorgänge voneinander elementar. So dürfen keinesfalls Daten „verloren gehen“. Aber auch die Mehrfachlieferung unveränderter Daten kann zumindest lästig sein und unnötig Zeit und Ressourcen verschwenden. Optimal wären hier absolut zuverlässige Varianten. Aber die gibt es nicht! Zumindest nicht „umsonst“, wobei hier nicht unbedingt Geld gemeint ist. So ist bspw. Asynchrones Change Data Capture (ACDC), Golden Gate oder praktisch jedes andere, auf Log-Informationen basierende Verfahren an die Vollständigkeit der Redo-Log Einträge gebunden – was im Quellsystem üblicherweise durch Supplemental Logging bzw. das Unterdrücken von NOLOGGING Operationen gewährleistet wird. Zudem gibt es für manche, Logfile basierte Verfahren Einschränkungen bei den Datentypen oder bei komprimierten Blöcken. Andere Methoden arbeiten mit Triggern oder sogar applikationsgesteuert – was per se schon einen deutlichen Overhead produziert und immer auch ein zusätzliches Risiko darstellt (Fehlerquelle für Betrieb des Quellsystems oder Unvollständigkeit des Inkrements). Sonstige Methoden sind zwar oft gut für Schnappschüsse, aber nicht für eine vollständige Erfassung aller Änderungen und Löschvorgänge geeignet.

Bleibt noch die Beurteilung im Bezug auf Near-Time Data Warehousing. Besonders die einfache und schnelle Ermittel- und Extrahierbarkeit von Änderungen ist hierbei essentiell wichtig – und natürlich deren Transport.

Transport

Nun müssen die extrahierten Daten auch zum Data Warehouse gebracht werden. Neben den einfachen Varianten wie Dateitransfer per Netzwerk-Share, FTP, SCP und Verwandtschaft oder dem allgegenwärtigen Datenbank-Link sind auch andere Varianten – oft im unmittelbaren Zusammenhang mit der gewählten Extraktionsmethode – möglich. So dürfen neben Queueing oder Logfile Transfer auch der Transportable Tablespace oder die Befüllung via extern arbeitendem ETL Tool (bspw. Oracle Data Integrator, Powercenter oder Talend) nicht vergessen werden. Und nicht zuletzt könnten manche Daten auch per Enterprise Service Bus (ESB) oder einem anderen Vertreter der SOA Familie ins Data Warehouse geschoben oder gezogen werden.

Je nach Bandbreite des Netzwerkes zwischen Quell- und DWH System ist dazu auch noch eine Kompression der übertragenen Daten zu berücksichtigen. Bei großen Datenmengen und schwachem Netz kann man sich schon mal ein paar Stunden Laufzeit sparen.

Laden

Wie schon erwähnt lassen sich CSV oder FixedLength Dateien am einfachsten und schnellsten per SQL*Loader oder External Table in die Datenbank laden. Aber was ist mit XML Dateien oder anderen, „absurden“ Dateistrukturen? Zunächst mal: Der große Overhead und die im Vergleich zu flachen Dateien höhere Komplexität von XML sind der Verarbeitungsperformance nicht gerade zuträglich. Man sollte diese Variante wohl besser nicht für 100 Millionen täglich zu verarbeitende Call Data Records („Anrufe“ in der TelCo Branche) nutzen. Für kleine und mittlere, vor allem aber zeitunkritische Datenmengen kann dieser Overhead jedoch durchaus akzeptabel sein, wenn man dafür beispielsweise eine bereits existierende, standardisierte Quellsystemschnittstelle nutzen darf. Zum Laden bieten sich hier native XML Funktionen der Datenbank an, um die XML Daten gleich beim Befüllen der Staging Area zu „relationalisieren“. Alternativ bringen die meisten ETL Werkzeuge entsprechende Funktionen mit. Wenn Sie jedoch die freie Wahl haben, ob Sie lieber mit flachen oder XML Daten arbeiten wollen, sollte die Wahl im DWH Umfeld nicht schwer fallen...

Bei manchen Methoden und Werkzeugen ist dagegen der Ladeprozess „fest eingebaut“. Ob Golden Gate, Transportable Tablespaces, Change Data Capture oder Datenbank-Link: Die Quelle für die Weiterverarbeitung innerhalb des Data Warehouse ist eine, durch das Tool befüllte Tabelle, eine View oder ein Link – auf jeden Fall etwas, dass sich per SQL SELECT einfach lesen lässt.

Wer …?

Welche Lösungen werden denn heute üblicherweise eingesetzt bzw. sind prinzipiell möglich?
Wir betrachten dabei vor und Nachteile zweier „fertiger“ Lösungen: Oracle Change Data Capture (CDC), das als Bestandteil der Oracle Database Enterprise Edition zahlreiche Varianten des Datenflusses zwischen zwei Oracle Datenbanken bietet und Oracle Golden Gate, das auch den Datenaustausch zwischen Datenbanken unterschiedlicher Hersteller unterstützt – und das nicht nur in eine Richtung, weswegen man es durchaus auch als Replikationswerkzeug einsetzen kann.
Darüber hinaus schauen wir uns auch eine Hand voll „manueller“ Methoden genauer an. Als Transportverfahren gehen wir dabei einfacherweise von einem Datenbank-Link oder einem FlatFile aus.

Alle betrachteten Methoden schöpfen im Prinzip aus dem gleichen Fundus von Möglichkeiten, den ich hier in vereinfachter Form kurz vorstellen möchte

Identifikationstechnik

- SELECT ohne Filter (Vollabzug)
- Redo/Archive-LogFile apply (Data Guard) mit und ohne supplemental Logging / LogMiner oder proprietären Änderungserkennungsmethoden
- Trigger/SCDC
- Änderungsspalte(n)
- Journaltabelle(n)
- ORA_ROWSCN

Extraktionstechnik

- Redo-/Archive LogFile Write
- SQL*Net/Database Link Select

- Exp/DataPump Export (auch f. Transportable Tablespaces)
- Flat File Write
- XML File Write

Transporttechnik

- Redo-/Archive LogFile Copy/Write
- SQL*Net/Database Link Select/Insert
- Transportable Tablespace (Copy oder “liegen lassen”)
- File Copy

Ladetechnik

- Redo-/Archive LogFile apply
- SQL*Net/Database Link Insert
- Exp/DataPump Import (auch f. Transportable Tablespaces)
- SQL*Loader/External Table

… kann was?

Einige der hier vorgestellten Extraktionsmethoden sind plattformunabhängig, andere funktionieren nur mit einer Oracle DB, manche sogar nur ab einer bestimmten Version. Gemeinsam ist allen Methoden aber, dass sie zumindest mit Oracle als Quell- und als DWH Plattform realisiert werden können und dass es sich ausschließlich um Werkzeuge und Methoden aus dem Oracle Portfolio handelt. Werkzeuge anderer Hersteller nutzen letztlich aber dieselben Methoden und haben somit auch ganz ähnliche Vor- und Nachteile.

Versuchen wir, aus dem bisher gesagten eine Übersicht der wichtigsten Kriterien zu destillieren. Diese Kriterien werden üblicherweise direkt aus den Anforderungen abgeleitet und definieren den Rahmen für die Auswahl des optimalen Verfahrens.

- Quellsystembelastung
- DWH Belastung
- Netzwerkbelastung
- Erkennen gelöschter Daten
- Erkennen aller Änderungen zwischen Extraktionen
- Invasives Verfahren (greift in Applikation ein, bspw. Trigger)
- Risiko für Unvollständigkeit
- Abhängigkeit von Applikationsänderungen im Quellsystem
- Abhängigkeit von LOGGING Operationen
- Unterstützt alle Daten- und Segmenttypen

Wichtig für eine Auswahl ist auch die Komplexität einer Lösung. Wobei uns hier eigentlich nur die damit verbundenen Aufwende interessieren und dabei Entwicklung, Wartung und Training einfließen sollten.

- Konfigurations- und Entwicklungsaufwand im Quellsystem
- Konfigurations- und Entwicklungsaufwand im DWH

- Operative Komplexität

Fehlt noch das Thema Performance. Damit ist die Gesamtperformance des Verfahrens, also die Laufzeit einer E+L Prozesskette inklusive eines ggf. nötigen Abgleichs mit dem DWH Bestand gemeint. Hier sollten wir für eine seriöse Beurteilung noch mindestens drei Subkriterien unterscheiden. Dabei ist die Eignung für Near-Time Data Warehousing aber im Wesentlichen von der Performance bei hoher Änderungsrate und weniger von der gesamten Datenmenge abhängig.

- Performance bei geringen Änderungsraten
- Performance bei hohen Änderungsraten
- Eignung für Near-/Real Time Warehousing

Dann spielen natürlich auch der Lizenzpreis und der Hardwarebedarf einer Lösung eine wichtige Rolle.

- Zusätzliche Lizenzkosten
- Zusätzliche Hardwarekosten

Ach ja, und eines der wichtigsten Kriterien überhaupt sollte man vielleicht auch nicht verschweigen: Die Kompatibilität. Welche Datenbanken und Versionen werden in Quelle und Ziel überhaupt unterstützt?

- Kompatibilität Quell- und Zielsystem

EL Matrix

Am Ende ergibt all das zusammengenommen folgende Matrix.

Natürlich handelt es sich auch bei dieser Tabelle nur um eine Vereinfachung. Nennen wir es Checkliste oder Leitfaden, der bei der ersten Orientierung hilft, die größten Fehler zu vermeiden. Auf keinen Fall aber kann diese Matrix eine eingehende Analyse ersetzen. Dafür sind die Kriterien zu allgemein formuliert und die einzelnen Möglichkeiten der Lösungen im Detail nicht darstellbar.

Methode	Change Data Capture					GoldenGate als Basis für CDC	Standby Datenbank Staging	Änderungsmarker			Vollbestand-abgleich
Kommentar	> 10.2.0.4 empfehlenswert (Stabilität)						immer Vollbe-stand mit DWH abzugleichen	Identifikation via "LastDate/Value" und Übertragung via DB Link oder FlatFiles (CSV)			
Lizenzen (Minimum)	DB SE 9.2	DB EE 10g					DB EE 10.2	Alle DBs ab 7.1		DB XE 10g	DB XE 10g
Variante	Synchron	Asynchron									
Subvariante		Autolog Online	Autolog Archive	Hotlog	Distributed Hotlog	GoldenGate Lizenz	Physical Standby	Change-Spalte	Journal-tabellen	ORA_ROWSCN	
Quellsystem-Belastung	hoch	minimal	minimal	hoch	mittel	mittel	minimal	mittel - hoch	hoch	gering	gering - mittel
Netzwerk Belastung	gering	mittel	mittel	gering	gering	gering	mittel	gering	gering	gering	hoch
DWH System-Belastung	gering	mittel	mittel	gering	mittel	mittel	mittel	gering	gering	gering	hoch
Konfig. & Entwicklungsaufwand Quellsystem	gering	gering	gering	gering	gering	gering	gering	App oder Trigger	App oder Trigger	minimal	kein
Konfig. & Entwicklungsaufwand DWH	gering	gering	gering	gering	gering	gering	gering	gering	gering	gering	mittel
Komplexität (operativ)	mittel	mittel	mittel	mittel	mittel	mittel	gering	gering	mittel	minimal	gering
DELETE Erkennung	ja	ja	ja	ja	ja	ja	ja	nein	ja	nein	ja
Erkennung von Datenänderungen zwischen Extraktionen	ja	ja	ja	ja	ja	ja	nein	nein	ja	nein	nein
Quelle & Staging DB müssen identisch sein (OS, HW, Patchlevel)	selbe DB	ja	ja	selbe DB	nein	nein / auch hetrogene DB-Unterstützung (MS SQL, DB/2, etc.)	nein, aber mind. 10.2 als Quelle	nein	nein	nein	nein
zus. Lizenzkosten (außer durch Systembelastung)	keine	zweite DB	zweite DB	keine	keine	GoldenGate Lizenz!	nur wenn nicht auf DWH	keine	keine	keine	keine
zus. HW-Kosten (außer durch Systembelastung)	keine	ggf. für zweite DB	ggf. für zweite DB	keine	keine	keine	nur wenn nicht auf DWH	keine	keine	keine	keine
Invasives Verfahren (App.-Risiko)	ja (Trigger)	nein	nein	nein	nein	nein	nein	ja (Trigger oder Appänderung)	ja (Trigger oder Appänderung)	nein	nein
Datenverlust bei Nologging DML im Quellsystem	nein	ja	ja	ja	ja	ja	ja	nein	nein	nein	nein
Konfiguration / Wartungsaufwand	gering	mittel	mittel	mittel	mittel	mittel	mittel	gering	gering	gering	gering
Datenverlust-Risiko bei App.- Änderung	mittel	mittel	mittel	mittel	mittel	mittel	keine	hoch	hoch	gering	keine
Unterstützung aller Daten und Segmenttypen	ja	nein	nein	nein	nein	nein	ja	ja	ja	ja	ja
Relative Performance bei geringer Änderungsrate	hoch	hoch	hoch	hoch	hoch	hoch	mittel	hoch	hoch	hoch	gering
Relative Performance bei hoher Änderungsrate	gering	gering	mittel	gering	mittel	mittel	hoch	hoch	hoch	hoch	hoch
Real/Near-Time prinzipiell implementierbar?	RealTime (Tx) möglich	Zeitnah	Zeitnähe konfigurierbar	Zeitnah	Zeitnähe konfigurierbar	Zeitnähe konfigurierbar	Zeitnähe konfigurierbar	RealTime (Tx) möglich	RealTime (Tx) möglich	Zeitnähe implementierbar	Abh. von Tabellengröße ungeeignet

Abbildung 2: Methoden-Kriterien Matrix für Extraction & Load im Oracle Data Warehouse

Ich wünschen Ihnen eine gute Hand bei der Wahl der Methoden und Werkzeuge und viel Erfolg im Projekt.

Kontaktadresse:

Peter Welker
Trivadis GmbH
Industriestrasse 4
D-70565 Stuttgart

Telefon: +49 (0) 162-295 96 81
Fax: +49 (0) 711-90 36 32 59
E-Mail peter.welker@trivadis.com
Internet: www.trivadis.com