puting, ergibt sich eine schnelle und zuverlässige High-End-Transaktionsverarbeitung.

Fazit

Zusammenfassend lässt sich sagen:

- Coherence ist eine elegante und praktikable Lösung für Caching
- Coherence ist keine klassische persistierende Datenbank, sondern ein Zwischenspeicher (Cache)
- Es gibt umfangreiche Möglichkeiten, an Coherence anzudocken
- Coherence entlastet das Backend und beschleunigt die Zugriffe
- Dies führt zu weniger Wartezeit durch höhere Kapazitäten

Gabriele Jäger
gabriele.jaeger@oracle.com

Michael Fuhr
michael.fuhr@oracle.com

OWB – Wie finde ich den richtigen Nachfolger?

Michael Klose, CGI Deutschland Ltd. & Co. KG

Der Warehouse Builder wird von Oracle nicht weiterentwickelt und der Standard Support endet in naher Zukunft. Gerade Kunden, die die Datenbank-Version 12*c* nicht installieren wollen, sind gezwungen, sich nach Alternativen umzusehen.

Oracle stellt mit dem Data Integrator zwar einen würdigen Nachfolger bereit; dieser ist allerdings mit zusätzlichen Lizenzkosten verbunden. Der Artikel zeigt verschiedene Möglichkeiten des Umstiegs auf ein anderes ETL-Tool auf. Sie werden anhand von Praxiserfahrungen aus einer ETL-Toolauswahl dargestellt. Darüber hinaus werden die Vor- und Nachteile der verschiedenen Tools sowie grundsätzliche Veränderungen in der Entwicklungstätigkeit und Architektur aufgezeigt. Die Schwerpunkte liegen bei Oracle Data Integrator, Informatica Powercenter, Talend, PL/SQL und einer Empfehlung für die Vorgehensweise bei der Tool-Auswahl.

Grundsätzliches

Bei den ETL-Tools gibt es durch die große Anzahl von Entwicklungswerkzeugen gerade aus Architektur-Sicht unterschiedliche Ansätze der Hersteller. Grundsätzlich unterscheidet man zwischen ETL (Extract – Transform – Load) und ELT (Extract – Load – Transform). Beim ETL-Prozess werden die Daten aus den Quellsystemen beispielsweise als Flat Files extrahiert, im Anschluss in einer ETL-Engine transformiert, um letztendlich final in die Zieldatenbank (Data Warehouse) geladen zu werden. Im Gegensatz dazu laden ELT-basierte Werkzeuge die Daten zuerst in die Zieldatenbank, um dort dann die Transformationen durchzuführen. Ein weiterer Unterschied liegt in der Programmiersprache des generierten Codes der Transformationsprozesse.

OWB – Stärken und Schwächen

Gerade für die in der Oracle-Programmierung kompetenten Entwickler stellt die Generierung von PL/SQL-Code eine klare Stärke dar. Der Code ist komplett nachvollziehbar und für die jeweilige Datenbank-Version optimiert. Nahezu alle Datenbank-Features können verwendet werden und die Ausführung erfolgt direkt auf der Datenbank (Push-Down). Die hohe Integration in die Oracle-Produktpalette und die einfache Installation innerhalb der Datenbank bedingen keine zusätzlichen Backup-Aufwände. Das vollständige, teilweise sogar Produkt-übergreifende Metadaten-Repository (OBI) ermöglicht detaillierte Data-Lineage-Analysen und unterstützt das Deployment auf andere Umgebungen. Der Funktionsumfang im ETL-Bereich umfasst alle wichtigen Komponenten wie Mappings, Workflows, Fehlerprotokollierung etc. Eine nicht zu vernachlässigende Stärke ist auch, dass der Oracle Warehouse

Builder in der Standardvariante ein Bestandteil der Datenbank-Lizenz war.

Leider bestehen neben den vielen Vorteilen auch gravierende funktionale Mängel. Gerade in sehr heterogenen System-Landschaften geriet der OWB schnell an seine Grenzen. Für den Datenabzug aus Nicht-Oracle-Datenbanken konnten noch Gateways oder Heterogeneous Services eingesetzt werden, allerdings war als Ziel-Datenbank nur Oracle sinnvoll einsetzbar. Ebenso kann nur ELT-basiert gearbeitet werden, was immer eine Integration der Daten in die Oracle-Datenbank bedingte. Sicherlich traf das den Großteil der Anwendungsfälle für ein Data Warehouse, allerdings werden ETL-Tools heutzutage vielschichtiger eingesetzt.

Mankos gibt es auch im Bereich des generierten PL/SQL-Codes. Dieser war zwar lesbar, es ist jedoch dringend davon abzuraten, den Code nach der Generierung zu verändern, da diese Veränderungen bei der nächsten Generierung verloren wären.

Zusätzlich zur lizenzfreien Variante konnte man sich weitere Funktionalitäten innerhalb einer Enterprise Version lizenzieren. Funktionen wie Lade-Reihenfolge für Tabellen, Schleifen im Workflow, Variablen-Übergabe oder das automatisierte Befüllen von Slowly Changing Dimensions sind extra zu lizenzieren oder mit mehr oder weniger aufwändigen Workarounds zu lösen. Das Thema „Versionierung" war jahrelang ein Enhancement Request bei Oracle, wurde jedoch nie realisiert.

ETL-Tools und wichtige Funktionalitäten im Überblick

Für eine Tool-Auswahl wird auf jeden Fall die Heranziehung von Analysten-Bewertungen (Gartner, Forrester, Barc) als Basis empfohlen, da eine Betrachtung aller marktrelevanten ETL-Tools einen sehr hohen Aufwand bedeutet. Will man die Auswahl von Beginn an stark einschränken, kommen sicherlich der Oracle Data Integrator als Produkt-Nachfolger, Informatica Powercenter als Marktführer und möglicherweise Talend als Open-Source ETL-Tool infrage.

Bezüglich der Anforderungen an moderne ETL-Tools müssen diese die klassischen Funktionalitäten enthalten. Eine große Funktionsbibliothek, Workflow-Steuerung, Wiederverwendbarkeit von Teilkomponenten, integrierte Versionsverwaltung, Deployment-Unterstützung, Metadaten-gestützte Entwicklung, Protokollierung und natürlich eine intuitive Bedienung sind selbstverständliche Bestandteile.

Im Zeitalter von Big Data und Self-Service-BI reichen diese Funktionen allein nicht aus. Hinzu kommen Anforderungen an die Integration unstrukturierter Daten, (Near-)Real-Time-Fähigkeiten, Anbindung von heterogenen Umgebungen, Change-Data-Capture-Mechanismen auf den Quellsystemen, Generierung von Mappings und Workflows, Skalierbarkeit und vor allem, vom Gesetzgeber teilweise sogar vorgeschrieben, die komplette Nachvollziehbarkeit von Datenveränderungen.

Oracle Data Integrator

Der Oracle Data Integrator (ODI) wird von Oracle als Nachfolger des Warehouse Builder (OWB) positioniert. In der aktuellen Version ist ein Migrationstool mitgeliefert, um je nach Komplexität der Mappings 70 bis 80 Prozent automatisiert zu migrieren. Der ODI ist ein ELT-Werkzeug und verwendet die Datenbank als ETL-Engine. ODI-Neueinsteiger bezeichnen den ODI-Agent gerne als „ETL-Engine". Dabei handelt es sich um einen Java-Prozess, der für die Ausführung der generierten Statements auf den Datenbanken zuständig ist, selbst aber keine Transformationen durchführt.

Eine weitere Kern-Komponente sind die Knowledge-Module, die für die verschiedenen Datenbanken entsprechende SQL-Statements generieren, um die Transformations- und Ladeprozesse abzubilden. Eine eigene ETL-Engine besitzt ODI hingegen nicht und kann somit nur die Funktionsbibliotheken der verwendeten Datenbanken nutzen. Der Benutzer kann die Knowledge-Module an die entsprechenden Anforderungen anpassen. Durch die Generierung von SQL-Statements bleibt der Code lesbar und kann beispielsweise für Tuning-Zwecke gut nachvollzogen werden.

Weitere wichtige Bestandteile sind Workflow und Load Plans. Diese Komponenten realisieren die Lade- und Ablaufsteuerung. Sie sind im Vergleich zum OWB wesentlich mächtiger. Dies zeigt sich vor allem im Bereich der Wiederaufsetzbarkeit von Ladeprozessen nach Abbrüchen.

Auch in der Versionierung und Paketierung für das Deployment gibt es nennenswerte Verbesserungen. Im ODI besteht die Möglichkeit, jede einzelne Komponente zu versionieren und in Deployment-Pakete zusammenzufassen. Der Funktionsumfang von Versionierungswerkzeugen wie Subversion wird damit zwar nicht erreicht, allerdings ist ein sinnvolles Arbeiten mit Versionierung möglich.

Im Gegensatz zum OWB lässt sich eine Vielzahl von Datenbanken als Quell- und Zielsystem anbinden. Die Verbindung zur Datenbank wird über den ODI-Agent und entsprechende JDBC-Treiber hergestellt. Für nahezu alle namhaften Datenbank-Hersteller sind umfangreiche Best-Practice-Knowledge-Module enthalten. Dies hat zur Folge, dass der ODI auch in sehr heterogenen System-Landschaften im Gegensatz zum OWB sehr gut eingesetzt werden kann.

Informatica Powercenter

Informatica Powercenter (IPC) gehört zur Kategorie der ETL-Tools und ist sicher eines der mächtigsten und umfangreichsten Werkzeuge im ETL-Bereich, was Analysten regelmäßig bestätigen. Der Einsatz einer eigenen ETL-Engine führt zu einer Unabhängigkeit gegenüber der Funktionsbibliothek der Datenbank. Allerdings benötigt diese Engine einen (im Normalfall) eigenständigen Server zur Ausführung der Transformationsprozesse. Dadurch ist es möglich, Daten aus verschiedenen Quellen zu extrahieren, in die ETL-Engine zu laden, Transformationen, Joins und Aggregationen durchzuführen sowie die Daten final in die Zieldatenbank zu schreiben. In der klassischen Datawarehouse-Layer-Architektur (Stage/EDWH/Data Marts) ist es in diesem Fall allerdings auch notwendig, die Daten beim Transport durch die verschiedenen Layer aus dem Warehouse zu extrahieren und nach den Transformationen wieder in dieses zu laden (etwa EDWH nach Data Mart).

Informatica bietet für Powercenter zwar eine sogenannte „Push-Down-Funktion" an, um möglichst viele Tätigkeiten auf die Datenbank auszulagern, allerdings kann dort nicht alles ausgeführt werden. Diese Funktionalität war in der Vergangenheit mit zusätzlichen Kosten verbunden und wurde deswegen vom Kunden häufig nicht eingesetzt. Um beispielsweise bei Delta-Verarbeitung die Performance beim Laden zwischen den DWH-Layern zu verbessern, werden häufig die „Source Qualifier" (ein Informatica-Objekt, welches das SELECT-Statement enthält) manuell

mit Joins und Filtern angepasst. Dies führt zu einem Verlust der Data-Lineage-Funktionalität in diesem Bereich, ist aber häufig die einzige Optimierungsmöglichkeit.

IPC generiert keinen lesbaren Code, der optimiert und visualisiert werden kann, und stellt somit eine „Black Box" dar. Die Oberfläche selbst erinnert stark an den OWB und aus der Erfahrung zeigt sich, dass Mitarbeiter mit OWB-Kenntnissen sehr schnell beginnen können, Mappings in IPC zu entwickeln.

Zur Ausführung und Prozess-Steuerung ist eine Workflow-Komponente verfügbar. Ohne ein externes Workflow/Scheduling-Tool wie CTRL-M ist die Vorgehensweise gerade beim Verschachteln von Workflows eher schlecht gelöst. Die kostenpflichtige Option „Team Based Development" ermöglicht die Versionierung sowie ein Check-In/Out ähnlich Subversion. IPC besitzt viele (lizenzpflichtige) Konnektoren zu Datenbanken, die bei den großen Datenbank-Herstellern sogar ein Change Data Capture für den Datenabzug unterstützen.

Talend Enterprise Data Integration

Talend bietet abhängig von den Anforderungen verschiedene Ausbaustufen seiner Data Integration Suite an. Um das Leistungsspektrum des OWB abzubilden, wird hier die „Enterprise Data Integration"-Variante betrachtet. Talend gehört ebenso wie Informatica zur Kategorie der ETL-Tools. Als ETL-Engine dient hier letztendlich die Java-Runtime-Engine.

Mappings werden wie bei den vorherigen Tools grafisch entwickelt und ausführbarer lesbarer Java-Code generiert. Die Anbindung von Quell- und Zielsystemen erfolgt über JDBC-Treiber, die fast alle Datenbank-Herstellern anbieten. Dadurch eignet sich Talend für sehr heterogene Quell- und Zielsysteme.

Die Daten müssen zur Durchführung von Transformationen aus dem Data Warehouse extrahiert und in der Java-Runtime-Engine verarbeitet werden. Hierfür ist ein zusätzlicher ETL-Server empfehlenswert, der vor allem in den Bereichen „CPU" und „Memory" nicht zu klein ausgestattet sein sollte. Der Push-Down von Datenbank-Abfragen ist möglich, allerdings wird dies nicht für alle Datenbanken angeboten.

Die mitgelieferte Funktionsbibliothek, die out oft he box in einer grafischen Entwicklung verwendet werden kann, ist nicht sehr umfangreich. Dies hat zur Folge, dass die Transformationen in Java programmiert werden müssen. Hier steht wiederum der komplette Funktionsumfang von Java zur Verfügung, sodass auch weitere Libraries eingebunden werden können. Da es sich um reinen Java-Code handelt, bettet sich Talend optimal in Versionierungssoftware wie Subversion ein.

Oracle PL/SQL

Sicherlich nimmt Oracle PL/SQL für den ETL-Prozess bei einer Tool-Betrachtung eine Nebenrolle ein, nichtsdestotrotz haben viele Unternehmen mit Oracle-Datenbanken diese Variante erfolgreich im Einsatz. Im Vergleich zu ETL-Tools wie Informatica oder Talend sind PL/SQL und SQL wesentlich näher am Ergebnis der Mapping-Generierung des OWB. Die großen Vorteile der Tools sind grafische Entwicklung, Metadaten-Repository, automatische Protokollierung und Data Lineage. Auf den ersten Blick stellen diese nicht zu vernachlässigende Komponenten dar, allerdings können einige auf andere Weise in einem Oracle-Datenbank- und im PL/SQL-Umfeld alternativ abgebildet werden.

Die grafische Entwicklungsoberfläche ist sicherlich nicht durch PL/SQL zu ersetzen. Eine Variante, um SQL-Statements zu generieren, stellt beispielsweise der Query Builder dar. Allerdings werden nur wenige Entwickler diese Alternative einsetzen wollen.

Beim Metadaten-Repository und der Data Lineage ist die Diskrepanz lange nicht so groß. Letztendlich stellt das Data Dictionary der Datenbank eine Art „Metadaten Repository" dar. Auf Datenbankobjekt-Ebene können hier viele Informationen hinterlegt werden. Für die Transformations-Metadaten bietet es sich an, ein eigenes Mini-Repository anzulegen. Ein solches steht in der Oracle-DWH-Community kostenfrei zur Verfügung. Damit lässt sich eine Data-Lineage-Analyse auf einfachem Weg aufsetzen; andernfalls bietet die Datenbank selbst Möglichkeiten, die Abhängigkeiten zwischen verschiedenen Objekten auszuwerten.

Ein wichtiger Bestandteil jedes ETL-Prozesses ist die Protokollierung der Laufzeit-Informationen. In diesem Bereich existieren fertige Logging-Frameworks, die teilweise kostenfrei sind. Alternativ dazu können eigene Frameworks in PL/SQL entwickelt und auf einfachem Wege eingebunden werden.

Eine weitere wichtige Komponente stellt die Ablaufsteuerung dar. Gerade in Anbetracht der Parallelisierung von Ladeprozessen sind die Möglichkeiten von PL/SQL doch sehr begrenzt. Für diesen Zweck kann der Oracle Enterprise Manager verwendet werden. Dort lassen sich sowohl Datenbank- als auch Betriebssystem-Jobs definieren. Eine Parallelisierung der Prozesse und die Abbildung von Abhängigkeiten unter den Prozessen ist möglich.

Viele OWB-Entwickler verfügen über PL/SQL-Kenntnisse und somit wäre ein einfacher Umstieg − was die Entwicklung betrifft − möglich. Bei bestehenden OWB-Implementierungen könnten die „Insert As Select"-Statements über OMB+ als Script generiert werden und somit die Basis für die Migration darstellen. Selbst Row-Based-Mappings können überführt werden. Voraussetzung dafür ist allerdings, dass man die OWB-Komponenten (Logging, Initialisierung etc.) vorher entfernt.

Die ETL-Tool-Auswahl

Der wichtigste Punkt, um das richtige ETL-Tool beziehungsweise einen OWB-Nachfolger auszuwählen, ist eine genaue und möglichst detaillierte Anforderungserfassung. Diese sollte die häufig genutzten OWB-Funktionalitäten identifizieren, vermisste sowie benötigte Funktionen enthalten sowie um zusätzliche, möglicherweise bereits bekannte Funktionalitäten für die Zukunft (Big Data) ergänzt werden.

Wer wählt das Tool aus? Diese Frage ist sehr wichtig, da Entwickler, Architekten und auch der Betrieb häufig unterschiedliche Betrachtungen, gerade im Bereich der Gewichtung von Funktionalitäten, haben. Sobald das Team zusammengestellt und die Anforderungen erfasst sind, erfolgt die Erstellung des Anforderungskatalogs, der dann an die verschiedenen Anbieter zur Beantwortung versendet werden kann. Wichtig sind hier die Benennung von K.O.-Kriterien (wie Betriebssystem-Unterstützung) sowie die Festlegung der Gewichtung der einzelnen Anforderungen nach ihrer Wichtigkeit. Dies sollte unbedingt vor der Auswertung erfolgen, um den Prozess zu beschleunigen.

Liegen die Antworten der möglichen Hersteller aus der Long List vor, kann über den Katalog die Reduzierung der möglichen Kandidaten erfolgen. Im weiteren Verlauf lädt man wenige Kandidaten für Präsentationstermine (Short List) ein. Um eine Vergleichbarkeit zu gewährleisten, sollte den Herstellern ein fester Fahrplan für die Präsentation der angeforderten

Features zugeteilt werden. Anhand der Ergebnisse reduziert sich die Anzahl der Hersteller weiter, sodass die finalen Kandidaten zu einem intensiven, mehrtägigen „Proof of Technology"-Workshop eingeladen werden können. In jedem dieser Schritte gilt die Prämisse: „Keine Angst, Kandidaten auszusortieren."

Im „Proof of Technology"-Workshop werden final die gestellten Anforderungen nochmals live geprüft. Es empfiehlt sich, während der gesamten Dauer Spezialisten aus dem eigenen Unternehmen den Workshop begleiten zu lassen. Dies stellt sicher, dass auch im Nachgang bereits das Wissen darüber existiert, wie die gestellten Anforderungen umgesetzt werden können.

Fazit

Der Artikel zeigt anhand von Beispielen die verschiedenen Möglichkeiten des Umstiegs auf ein anderes ETL-Tool. Da jedes Tool seine Vor- und Nachteile hat, ist es umso wichtiger, die Anforderungen, individuell auf das eigene Umfeld betrachtet, festzulegen und in einem Tool-Auswahlprozess zu bewerten.

Michael Klose
michael.klose@cgi.com

OWB ohne OWB: Wie rette ich meine ETL-Sourcen nach 12c R2?

Sven Bosinger, its-people GmbH

Oracle hat angekündigt, den Oracle Warehouse Builder (OWB) ab der Datenbank-Version 12*c* R2 nicht mehr zu unterstützen. Dies hat zur Folge, dass alle ETL-Prozesse, die mit dem OWB erstellt wurden, in ein neues Werkzeug migriert werden müssen. Oracle empfiehlt hier den Oracle Data Integrator (ODI). Doch ist eine Migration immer notwendig?

Der Artikel stellt eine Möglichkeit vor, die mit dem OWB erzeugten ETLs ohne den OWB und damit potenziell auch unter 12*c* R2 weiter zu nutzen. Dazu wurde ein ETL-Workframe geschaffen, der die fehlenden OWB-Runtime-Komponenten ersetzt und so eine Lauffähigkeit der Sourcen ohne Installation des OWB ermöglicht.

Motivation

Im Januar 2010 und später noch einmal konkretisiert im Oktober 2013 hat Oracle folgendes Statement of Direction veröffentlicht: „No major enhancements are planned for Oracle Warehouse Builder beyond the OWB 11.2 release. OWB 11.2 continues to be available and supported by Oracle, and patches and bug fixes will continue to be offered at regular intervals. Oracle will continue to support OWB 11.2 for the full lifetime of Oracle Database 11*g* in accordance with Oracle's Lifetime Support Policies for Oracle Database releases. Future database releases beyond Oracle Database 12*c* Release 1 will not be certified with OWB 11.2." Daraus ergeben sich für die Nutzer des OWB folgende Konsequenzen:

- Wer OWB einsetzt, muss sich zeitnah Gedanken über eine Migrationsstrategie machen.
- Die Datenbank 12*c* R2 ist für Ende 2015 / Anfang 2016 angekündigt. Wer diese einsetzen möchte, muss zuvor alle OWB-Mappings migrieren.
- Der Premium-Support für 12*c* R1 endet im Juli 2018, der Extended Support im Juli 2021. Ein Betrieb über den Juli 2018 hinaus zieht zusätzliche Support-Kosten nach sich. Nach dem Juli 2021 ist ein Support in der Regel nicht mehr möglich.
- Das von Oracle präferierte Nachfolgeprodukt des OWB ist der Oracle Data Integrator (ODI). Eine Migration dorthin ist aufwändig und zieht Kosten nach sich. Der ODI deckt nicht alle Funktionalitäten des OWB ab. Es ist davon auszugehen, dass bei einer durchschnittlichen DWH-Datenbewirtschaftung maximal 90 Prozent der ETL-Sourcen „1:1" migriert werden können. Für die restlichen 10 Prozent ist eine zum Teil aufwändige Neuentwicklung notwendig (Erfahrungswerte).
- Der ODI ist nicht wie der OWB (Standard Edition) in den Lizenzkosten der Datenbank enthalten, sondern muss zusätzlich lizenziert werden.