

Performance by Design – Wie werden performante ETL-Prozesse erstellt?

Reinhard Mense
ARETO Consulting
Bergisch Gladbach

Schlüsselworte:

DWH, Data Warehouse, ETL-Prozesse, Performance, Laufzeiten, Partitionierung, Materialized Views, Statistiken

Einleitung

Die ETL-Prozesse für die Befüllung eines Data Warehouse Systems (DWH) müssen in einem begrenzten Zeitfenster große Datenmengen verarbeiten. Man braucht verlässliche, d. h. im Idealfall konstante Laufzeiten für die ETL-Prozesse, damit sicher gestellt ist, dass das Zeitfenster auch zukünftig ausreicht und man keine bösen Überraschungen erlebt.

Für die Entwicklung performanter ETL-Prozesse besteht die Schwierigkeit darin, dass zwar die aus den Quellsystemen zu verarbeitende Datenmenge sich in der Regel nur geringfügig ändert, die historische Datenmenge im DWH jedoch deutlich und stetig zunimmt. Um nahezu konstante Laufzeiten der ETL-Prozesse zu erreichen, müssen die Laufzeiten unabhängig von der historischen Datenmenge im DWH sein. Die Erstellung performanter ETL-Prozesse mit konstanten Laufzeiten stellt somit eine der großen Herausforderungen bei der Entwicklung eines DWH dar.

Die Oracle Datenbank verfügt über eine Vielzahl von Technologien, die die Verarbeitung von großen Datenmengen in kurzer Zeit ermöglichen. Werden diese schon beim Entwurf eines Data Warehouse berücksichtigt und richtig eingesetzt, können ETL-Prozesse mit annähernd konstanten Laufzeiten erzielt werden.

Ein Beispiel aus der Praxis

Die Bedeutung eines gut durchdachten technischen DWH-Entwurfs wird bei der Betrachtung eines negativen Beispiels aus der Praxis deutlich:

Ein Kunde hat für die Beladung des DWH ein Zeitfenster von lediglich 120 Minuten zur Verfügung. Die Beladung findet monatlich statt. Pro Monat werden etwa 6 bis 7 Millionen neue Datensätze verarbeitet. Die historische Datenmenge im DWH soll später einmal mindestens zehn Jahre umfassen.

Während die erste Beladung nur wenige Minuten in Anspruch nahm und damit noch deutlich innerhalb des zur Verfügung stehenden Zeitfensters lag, stellte der Kunde schon nach wenigen Monaten eine stetige Zunahme der Laufzeiten der ETL-Prozesse fest. Bereits nach 12 Monaten bzw. 12 Beladungen drohte das Zeitfenster nicht mehr auszureichen.

Um ein Überschreiten des Zeitfensters zu verhindern, beschloss der Kunde für das DWH eine leistungsfähigere Hardware einzusetzen. Durch den Einsatz der besseren Hardware reduzierten sich die Laufzeiten um fast die Hälfte, so dass das zur Verfügung stehende Zeitfenster zunächst wieder

eingehalten werden konnte. Im weiteren Zeitverlauf stiegen die Laufzeiten aber weiter kontinuierlich an. Nach 21 Monaten wurde schließlich trotz der leistungsfähigeren Hardware das Zeitfenster überschritten, so dass keine dauerhafte Lösung erreicht wurde.

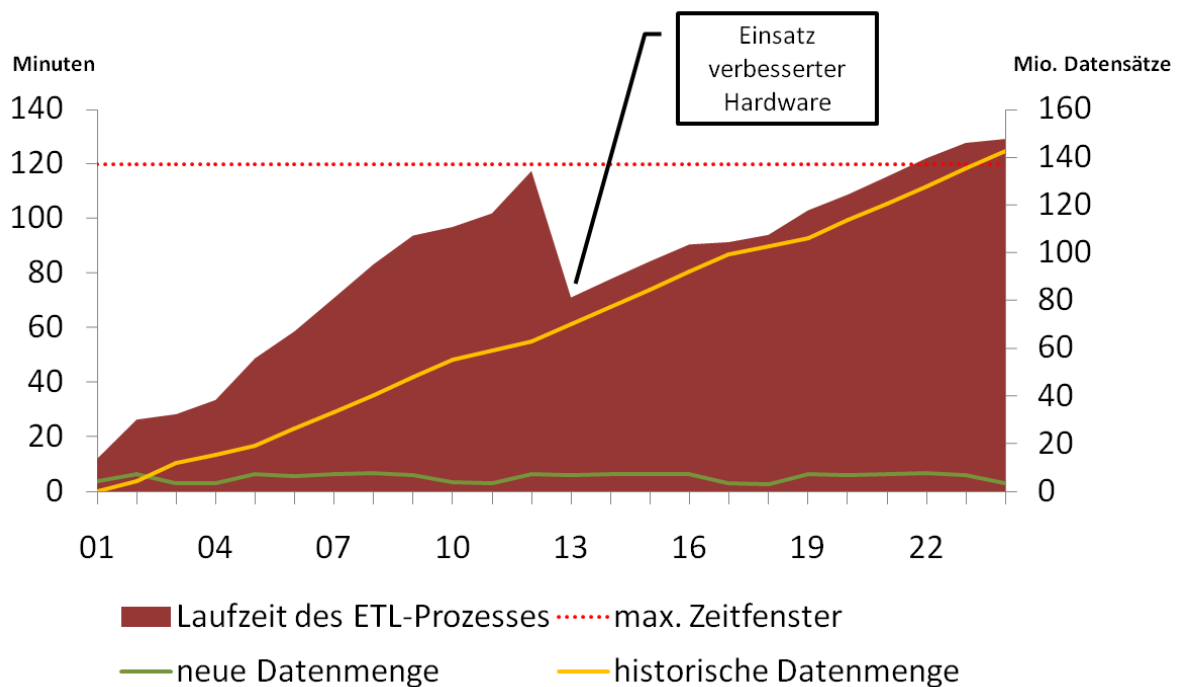


Abb. 1: Szenario mit steigenden Laufzeiten für den ETL-Prozess bei einem Kunden

In Abbildung 1 ist das Laufzeit-Szenario des Kunden grafisch dargestellt. Die zunehmende Laufzeit des ETL-Prozesses kann dabei nicht mit der zu verarbeitenden Datenmenge (neue Datenmenge) erklärt werden, da diese nahezu konstant bleibt. Vergleicht man die Laufzeit des ETL-Prozesses jedoch mit der historischen Datenmenge, so fällt hier eine Korrelation auf.

Partitionierung als Schlüssel für konstante Laufzeiten

Als Ursache für das schlechte Laufzeitverhalten im obigen Beispiel konnte die nicht konsequente Nutzung der Partitionierung ermittelt werden. Fehlende Partitionierung oder in den ETL-Prozessen nicht richtig berücksichtigte Partitionierung führt dazu, dass eventuell beim Zugriff auf die Zieltabellen eines DWH unnötigerweise auch die historischen Datenmengen verarbeitet werden.

Werden alle Objekte im DWH, die historische Daten aufnehmen, partitioniert, kann durch gezielten Zugriff auf einzelne Partitionen (z. B. Truncate einer Partition) oder durch die automatische Partitionseeliminierung der Oracle Datenbank (Partition Pruning) ein umfassender Zugriff auf die historischen Daten vermieden werden. Als Partitionierungsschlüssel sollte dabei das Datum der historischen Daten gewählt werden.

Der ETL-Prozess muss im Wesentlichen Tabellen, Materialized Views und Indizes mit neuen Daten befüllen. Sowohl Tabellen und Materialized Views als auch Indizes können partitioniert werden. Indizes sollten dann stets als lokal definiert werden, damit sie automatisch analog zur Tabelle oder Materialized View, für die sie erstellt werden, partitioniert werden.

Historische Datenmengen sind im DWH in erster Linie in den Fakttabellen enthalten, so dass diese auf jeden Fall zu partitionieren sind. Werden vom ETL-Prozess neu in die Fakttable aufzunehmende Daten zunächst in einer Staging Tabelle geladen, können die Daten durch einen Austausch der Staging Tabelle mit einer Partition der Fakttable unabhängig von den historischen Daten in das DWH aufgenommen werden. Für die Staging Tabelle aufgebaute Indizes können beim Austausch der Partition als lokale Indizes der Fakttable übernommen werden. Damit ist auch der Aufbau der Indizes unabhängig vom historischen Datenvolumen der Fakttable.

Historisierte Dimensionen (SCD Typ 2) können bei einem unzureichenden Design mit zunehmender Datenmenge – ähnlich wie Fakttabellen – zu einer stetigen Verlangsamung der ETL-Prozesse führen. Dimensionen mit großen Datenmengen (z. B. mehrere Millionen Datensätze) und häufigen Änderungen sind in heutigen DWH-Systemen nicht mehr selten. Mit zunehmender Datenmenge benötigt der ETL-Prozess mehr Zeit, für die Aufnahme der Änderungen in die Dimension und für den Lookup auf die Dimension. Durch die Partitionierung der Dimension nach dem Gültigkeitsende (Expire Date) können Dimensionsänderungen gezielt für eine Partition durchgeführt werden. Beim Lookup auf die Dimension wird bei Angabe des Gültigkeitszeitraums das Partition Pruning genutzt. In beiden Fällen erfolgt die Verarbeitung im ETL-Prozess somit unabhängig von der historischen Datenmenge.

Refresh von Materialized Views

Materialized Views werden im DWH typischerweise für die Bereitstellung aggregierter Daten erstellt, um so die Laufzeit von Berichten und Abfragen zu verkürzen. Damit die aggregierten Daten stets den aktuellen Stand aufweisen ist vom ETL-Prozess ein Refresh der Materialized Views durchzuführen. Der Refresh kann dabei vollständig (Complete Refresh) oder inkrementell (Fast Refresh) erfolgen. Um konstante Laufzeiten für den Materialized Refresh zu erzielen, ist ein Complete Refresh zu vermeiden, da dabei die kompletten historischen Daten der Basistabellen gelesen werden müssen. Aber auch bei einem Fast Refresh kann ein unerwünschter Zugriff auf die historischen Daten erfolgen.

Materialized Views im DWH basieren in der Regel auf einer Fakttable und gegebenenfalls per Join verknüpfter Dimensionstabellen. Den einfachsten Fall stellt eine Materialized View dar, die nur auf eine Fakttable basiert. In diesem Fall wird die Aggregation der Daten durch das Weglassen der Fremdschlüssel für einzelne Dimensionen erreicht. Wird eine solche Materialized View analog zu der Fakttable partitioniert, kann der Refresh per Fast PCT durchgeführt werden. PCT steht dabei für Partition Change Tracking und bedeutet, dass die Datenbank anhand von internen Log-Informationen automatisch die Partitionen der Fakttable erkennt für die die Daten geändert wurden. (Wichtig: Dafür ist kein Materialized View Log auf der Fakttable notwendig!) Da die Materialized View analog zur Fakttable partitioniert ist, werden beim Refresh auch nur die betroffenen Partitionen geändert. Geht man davon aus, dass beim ETL-Prozess keine historischen Daten in der Fakttable geändert werden, werden beim Refresh immer nur die Daten einer Partition erneuert. Somit können konstante Laufzeiten für den Refresh einer solchen Materialized View erzielt werden.

Schwieriger wird es, wenn die Materialized View neben der Fakttable auch auf per Join verknüpfte Dimensionstabellen basiert. Grundsätzlich kann man in diesem Fall zwar einen Fast Refresh basierend auf Materialized View Logs einsetzen. Allerdings garantiert dies keine konstanten Laufzeiten, da beim Refresh in allen Partitionen der Materialized View nach den zu aktualisierenden Datensätzen gesucht wird. Der Fast Refresh hat aber, sofern die geänderte Datenmenge nicht zu groß ist, deutliche Performance-Vorteile gegenüber einem Complete Refresh.

Die Datenbank unterscheidet beim Refresh nicht, ob es sich bei den beteiligten Dimensionen um historisierte Dimensionen (SCD Typ 2) oder um nicht historisierte Dimensionen (SCD Typ 1) handelt. Sind ausschließlich historisierte Dimensionen beteiligt, kann man aber ein manuelles Refresh-Verfahren entwickeln, um konstante Laufzeiten für den Refresh zu erzielen. Geht man davon aus, dass der ETL-Prozess keine historische Daten verarbeitet, werden weder historische Daten in der Faktabelle noch in den historischen Dimensionen geändert. Damit ist beim Refresh der Materialized View auch lediglich die aktuelle Partition zu aktualisieren. Erstellt man die Materialized View auf einer Prebuilt Table, kann man diese im ETL-Prozess manuell mit einem Partitionsaustausch aktualisieren. Für die kurze Dauer des Partitionsaustauschs muss die Materialized View gelöscht werden und anschließend wieder neu erstellt werden. Da die Materialized View für eine Prebuilt Table erzeugt wird, bleibt beim Löschen der Materialized View die Tabelle erhalten und benötigt das neue Erstellen der Materialized View keine nennenswerte Zeit.

Aktualisieren der Statistiken

Um performante Abfragen zu ermöglichen, sind für alle Tabellen, Materialized Views und Indizes im Rahmen des ETL-Prozesses die Statistiken zu aktualisieren. Für partitionierte Objekte kann dabei zwischen globale und lokale Statistiken unterschieden werden. Während lokale Statistiken nur die Daten jeweils einer Partition beschreiben, beziehen sich die globalen Statistiken auf die Daten aller Partitionen.

Das Erstellen der lokalen Statistiken ist nicht problematisch, da diese immer nur gezielt für die geänderte Partition erzeugt werden und deshalb mit konstanter Laufzeit möglich ist. Die Erstellung der globalen Statistiken ist dagegen deutlich aufwendiger. Auf die globalen Statistiken kann aber auch nicht verzichtet werden, da sie, sobald bei Abfragen auf mehr als eine Partition zugegriffen wird, vom Optimizer der Datenbank benutzt werden.

Da für die Statistiken unter anderem die Anzahl unterschiedlicher Werte für einzelne Spalten ermittelt werden, ist für die Erstellung der globalen Statistiken immer der gesamte Zeitraum zu betrachten. Es sind also immer die Daten aller Partitionen in Betracht zu ziehen. Konstante Laufzeiten sind somit nicht möglich! Mit Oracle 11g gibt es aber einen Ansatz globale Statistiken inkrementell zu erzeugen. Dadurch kann die Laufzeit für das Erstellen der globalen Statistiken drastisch reduziert und auch die Steigerung der Laufzeiten bei zunehmender historischer Datenmenge deutlich gemindert werden.

Inkrementelle Statistiken stehen ab Oracle 11g Release 1 zur Verfügung. In Oracle 11g R1 kann man für einzelne Tabellen die Erstellung inkrementeller Statistiken mit Hilfe der Prozedur DBMS_STATS.SET_TABLE_PREFS aktivieren.

```
DBMS_STATS.SET_TABLE_PREFS ('SH', 'SALES', 'INCREMENTAL', 'TRUE');  
DBMS_STATS.GATHER_TABLE_STATS ('SH', 'SALES');
```

Diese Anweisung veranlasst die Datenbank zusätzliche Informationen in sogenannten Synopsis-Objekten zu speichern. Diese Synopsis-Objekte benötigen nur sehr wenig Speicher und ermöglichen die inkrementelle Erzeugung der Statistiken. Mit zunehmender historischer Datenmenge nimmt zwar auch die Größe der Synopsis-Objekte zu, jedoch weisen diese auch dann immer noch eine relativ geringe Größe auf. Da für das inkrementelle Erstellen der globalen Statistiken lediglich die relativ kleinen Synopsis-Objekte gelesen werden, fällt die Steigerung der Laufzeit für das Erstellen der globalen Statistiken deutlich geringer aus.

Fazit

In einem DWH-typischen Testszenario wurden die beschriebenen Verfahren getestet. Dabei wurden sowohl die Laufzeiten des für ein nicht partitioniertes Szenario entwickelten ETL-Prozesses als auch die Laufzeiten des für ein partitioniertes Szenario optimierten ETL-Prozesses ermittelt.

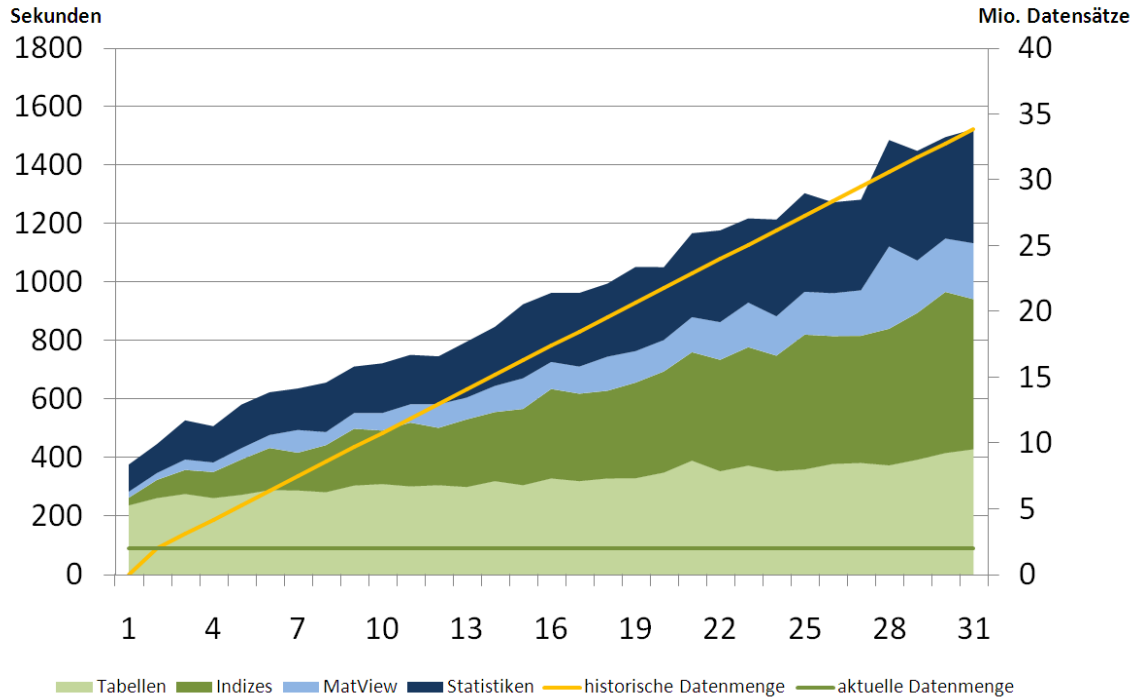


Abb. 2: Laufzeiten des ETL-Prozesse im nicht partitionierten Szenario

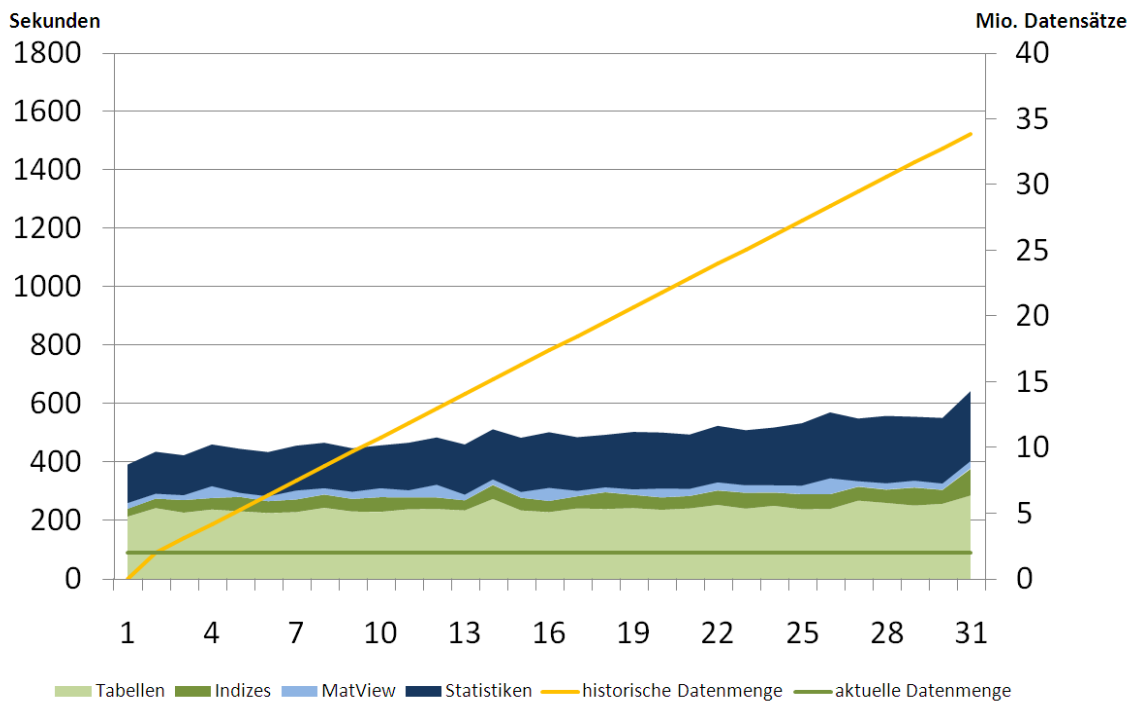


Abb. 3: Laufzeiten des ETL-Prozesse im partitionierten Szenario

Ein wesentlicher Erfolgsfaktor für ein Data Warehouse stellt dessen Performance dar. Dabei sind neben den Laufzeiten der Berichte und Abfragen auch die Laufzeiten der ETL-Prozesse von großer Bedeutung. Dabei ist zu beachten, dass das zunehmende historische Datenvolumen eines DWH großen Einfluss auf die Performance hat.

Durch ein kluges physikalisches Design kann eine gute Performance für ein DWH sichergestellt werden. Die Partitionierung erweist sich dabei als die essenzielle Technik für das Erzielen nahezu konstanter Laufzeiten für ETL-Prozesse. Datenbanktechniken, wie diese, müssen von Beginn an konsequent berücksichtigt werden. Bedenkt man, dass ein Großteil des Aufwands für ein DWH bei der Entwicklung der ETL-Prozesse entsteht, wird deutlich, dass die nachträgliche Behebung von technisch-strategischen Fehlern zu erheblichem Mehraufwand führt. Deshalb ist eine strategische Planung des technischen DWH-Designs erforderlich. Idealerweise greift man hier auf ein bewährtes Vorgehensmodell zurück. Statt auf nachträgliches Tuning zu vertrauen sollte das Motto für die Entwicklung „Performance by Design“ lauten.

Beim Kunden aus dem obigen Praxisbeispiel wurden durch die Einführung und konsequente Umsetzung eines Vorgehensmodells für ETL-Prozesse kürzere und konstante Laufzeiten der ETL-Prozesse erzielt (siehe Abbildung 4).

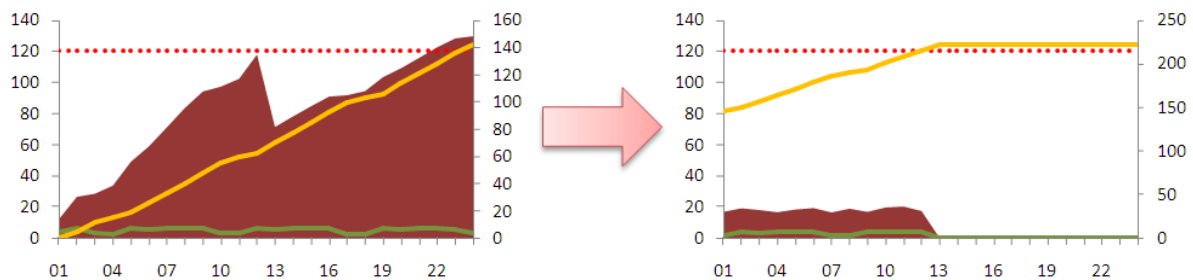


Abb. 4: Verbesserte Laufzeiten der ETL-Prozesse im Praxisbeispiel

Kontaktadresse:

Reinhard Mense

ARETO Consulting GmbH
Friedrich-Ebert-Str. 8
D-51429 Bergisch Gladbach

Telefon: +49 (0) 2204-300 878
Fax: +49 (0) 2204-300 879
E-Mail: reinhard.mense@areto-consulting.de
Internet: www.areto-consulting.de