

Kalamitäten mit Kardinalitäten

Randolf Geist
Unabhängiger Berater
Mannheim, Deutschland

Schlüsselworte

Effiziente Ausführungspläne, Cost-Based Optimizer, CBO, Grundlagen

Einleitung

Wenn es darum geht, effizientes SQL zu schreiben, gibt es einige wenige Schlüsselkonzepte, die zu verstehen sind. Obwohl es "kostenbasierter Optimizer" genannt wird, sind es eben nicht primär die Kosten, auf die wir schauen müssen, um zu verstehen, warum der Optimizer bestimmte Entscheidungen getroffen hat. In dieser Session werden diese Schlüsselkonzepte beleuchtet. Wir werden sehen, warum die Kardinalitäts- und Selektivitätsabschätzungen des Optimizers so elementar wichtig sind und welche Schlüsselkonzepte der Optimizer überraschenderweise derzeit (noch) nicht berücksichtigt. Dies bedeutet auch, dass wir manchmal wesentlich mehr über unsere Daten und Applikation wissen als der Optimizer und daher ihn in die richtige Richtung lenken müssen.

Drei entscheidende Fragen

Grundlegend gibt es bei der Suche nach einem effizienten Ausführungsplan drei entscheidende Fragen, die geklärt werden müssen:

1. Wie viele Zeilen werden erzeugt / Welche Datenmenge wird erzeugt?
2. Wie sind die zu suchenden Daten organisiert – eher zusammenhängend oder durcheinander gemischt?
3. Wie groß ist die Wahrscheinlichkeit, dass die zu suchenden Daten bereits im Cache sind?

Zwei Hauptstrategien

Warum sind es gerade diese Fragen, die so wichtig sind?

Weil es im Grunde zwei Strategien gibt, wie die Daten verarbeitet werden können:

1. Ein großer „Job“
Hier ist die Frage nach der Datenmenge entscheidend für die Abschätzung der Effizienz der Operation
2. Mehrere kleine „Jobs“
Hier ist die Frage nach wie viele Zeilen („wie oft muss der kleine Job ausgeführt werden?“) und wie sind die Daten organisiert („Wie groß ist der Aufwand pro Ausführung?“) entscheidend

Werden obige Fragen in diesem Zusammenhang richtig beantwortet, kann die passende Strategie ausgewählt werden. Umgekehrt ist es wahrscheinlich, dass bei falscher Beantwortung der Fragen eine unpassende Strategie ausgewählt wird, die weniger effizient sein kann.

Der Cost-Based Optimizer

Es ist mit Sicherheit nicht überraschend, dass der Cost-Based Optimizer (CBO) von Oracle versucht, die gleichen Fragen zu beantworten auf der Suche nach dem effizientesten Ausführungsplan.

Interessanterweise adressiert der CBO aber nicht alle drei Fragen im gleichen Umfang:

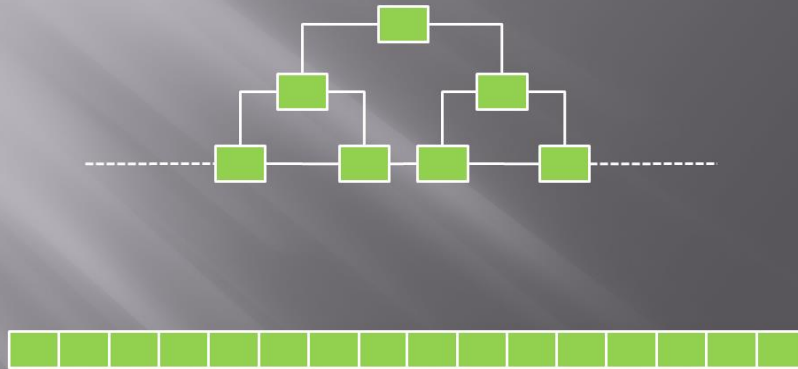
Die erste Frage nach der Anzahl der Zeilen und des Datenvolumens wird recht ausführlich behandelt, wobei vor allem die Abschätzung von Join-Operationen in Bezug auf Anzahl Zeilen und Datenvolumen auf recht groben Angaben (Statistiken) beruhen und daher potentiell ziemlich falsch sein können.



Schon bei der zweiten Frage nach der Organisation der Daten gibt es nur eine einzige Information, die dem CBO zur Verfügung steht: Der sogenannte „Clustering Factor“ von Indizes. Während der Clustering Factor eine in vielen Fällen ausreichende Information darstellt für die Entscheidung, wie auf eine einzelne Tabelle zugegriffen wird (Index oder Full Table Scan), gibt es je nach Ausführungsplan für Operationen, die verschiedene Datenquellen verknüpfen, überhaupt keine Information für den CBO, wie diese in Relation organisiert sind.

Dies festzustellen, wäre auch eine Aufgabe, die nach heutigem Stand der Technik wahrscheinlich zu viel Zeit und Ressourcen in Anspruch nehmen würde, da es so viele Möglichkeiten gibt, wie Datenquellen verknüpft werden können: Die Reihenfolge der Datenquellen kann unterschiedlich sein, und pro Datenquelle gibt es potentiell viele Zugriffsarten. Beides kann die Reihenfolge des Datenzugriffs verändern (zum Beispiel Full Table Scan C -> Index A -> Index B vs. Index C -> Index B -> Index A), was bedeutet, dass alle möglichen Kombinationen evaluiert werden müssten, um diese Information sinnvoll verarbeiten zu können.

HOW SCATTERED / CLUSTERED?



Die dritte Frage nach dem Caching von Daten wird derzeit vom CBO überhaupt nicht berücksichtigt.

Dies heisst aber, dass der CBO je nach Daten und Ausführungsplan nur über unzureichende Informationen verfügt, um diese Fragen korrekt beantworten zu können und daher leicht eine unpassende Strategie auswählen kann.

Weiterhin bedeutet dies bei ausreichendem Wissen über Ihre Daten und die Abfragen auf diese Daten, dass Sie unter Umständen diese Fragen wesentlich besser beantworten können wie der CBO und ihm daher helfen können, die passende Strategie auszuwählen, bzw. auch besser beurteilen können, ob der CBO eine passende Strategie ausgewählt hat.

Statistiken

Häufig sind auch noch zur heutigen Zeit Skripte und Implementierungen aktiv, die vor langer Zeit erstellt wurden. Insbesondere seit Oracle 11g und nochmals seit Oracle 12c sollten diese auf die verwendeten Aufrufe und Parameter hin überprüft werden. Der alte ANALYZE-Befehl sollte zur Erstellung von Statistiken schon lange nicht mehr verwendet werden. Aber auch die Aufrufe von DBMS_STATS sind oftmals veraltet – die Parameter ESTIMATE_PERCENT und METHOD_OPT spielen eine maßgebliche Rolle. Insbesondere die Auswirkung von ESTIMATE_PERCENT wird oftmals unterschätzt – nur bei der Verwendung des Defaults oder expliziter Angabe von DBMS_STATS.AUTO_SAMPLE_SIZE kann der neue „APPROXIMATE NDV“-Algorithmus zum Einsatz kommen, der in 11g die Basis-Statistiken, insbesondere den „Number of Distinct Values“-Wert von Spalten sehr akkurat über die gesamte Datenmenge erstellen kann, ohne dabei wie in früheren Versionen eine kostspielige Sortierung der analysierten Daten durchführen zu müssen. In 12c kommen weitere Features dazu, hier können mit dem neuen Modus hochqualitative Histogramme und auch neue Histogramm-Typen erstellt werden, die meisten davon in einem einzigen Durchlauf. Wird hier ESTIMATE_PERCENT mit einem alten Wert eingesetzt, stehen all diese neuen Features nicht zur Verfügung.

Auch der METHOD_OPT-Parameter hat maßgeblichen Einfluss auf die erstellten Statistiken, häufig wird noch „FOR ALL INDEXED COLUMNS“ verwendet. Dies ist sehr stark zu hinterfragen (bzw. war schon immer eine fragwürdige Option) und in den allermeisten Fällen durch den Default (FOR ALL COLUMNS SIZE AUTO) oder FOR ALL COLUMNS SIZE 1 zu ersetzen, da ansonsten die Basis-Statistiken für alle nicht indizierten Spalten nicht berechnet werden. Diese sind aber maßgeblich für die Volumen-Abschätzungen des Optimizers wichtig, was die Kostenberechnung für einen Hash Join signifikant verfälschen kann, wenn diese Basis-Statistiken nicht für alle in der Abfrage verwendeten Spalten vorliegen – hier zählen eben auch die Spalten dazu, die nur in der Projektion der Abfrage vorkommen und nicht Teil der WHERE-Klausel sind.

Histogramme

Insbesondere vor der Version 12c sind Histogramme nicht selten für schlechte Abschätzungen verantwortlich. Gerade der Histogramm-Typ „Height Balanced“ stellt viele Werteverteilungen von Spaltenwerten nur unzureichend dar. Dann sind Histogramme in Versionen vor 12c zumeist basierend auf sehr aggressivem Sampling (basierend auf nur ca. 5500 Zeilen unabhängig von der Anzahl der Zeilen in der Tabelle), was zusammen mit einigen Eigenarten bei der Kardinalitätsberechnung basierend auf Histogrammen zu merkwürdigen Effekten führen kann. Durch die Verwendung der Standard-Option des METHOD_OPT-Parameters (FOR ALL COLUMNS SIZE AUTO) werden potentiell zu häufig Histogramme erstellt, die dann eben insbesondere im Falle von „Height Balanced“-Histogrammen zu Problemen bei der Kardinalitätsabschätzung führen können.

Ab der Version 12c wurden in diesem Bereich endlich grundlegende Verbesserungen eingeführt. So können Histogramme jetzt im Rahmen der Statistik-Berechnung der Basis-Statistiken erstellt werden, es bedarf also kein zusätzlicher Zugriff mit Sampling, sondern die Histogramme basieren dann auf der gesamten analysierten Datenmenge. Dann gibt es neue Histogramm-Typen, Top Frequency und Hybrid. Für den Hybrid-Typ ist immer noch ein separater Zugriff auf die Daten pro Histogramm notwendig, und hier kommt auch wieder das aggressive Sampling zum Einsatz. Trotzdem sind auch diese Hybrid-Histogramme deutlich hilfreicher als die damit abgelösten „Height Balanced“-Histogramme. Für Frequency-Histogramme, die eine genaue Verteilung der Spaltenwerte repräsentieren, gibt es ab 12c die Möglichkeit, bis zu 2048 Werte abbilden zu können, im Gegensatz zum früheren Limit von 254.

Nur nochmals zur Klarstellung: Diese neuen Features von 11g und insbesondere 12c kommen nur zum Einsatz, wenn beim Aufruf von DBMS_STATS für den Parameter ESTIMATE_PERCENT der Default-Wert oder explizit DBMS_STATS.AUTO_SAMPLE_SIZE verwendet wird.

Verwendung falscher Datentypen

Nicht selten verwenden Applikationen nicht die korrekten Datentypen zur Speicherung von Datums- oder Zahlenwerten. Datumswerte werden dann häufig in Zahlen- oder Zeichenspalten abgelegt, Zahlenwerte eben auch in Zeichenspalten.

Je nach Art der verwendeten Filterkriterien kann dies zu groben Fehleinschätzungen des Optimizers führen, insbesondere bei Bereichsabfragen mittels größer und kleiner. Hier können als Workaround in manchen Fällen Histogramme abhelfen, aber grundsätzlich sollten möglichst die korrekten Datentypen zum Einsatz kommen, auch wenn der Oracle Optimizer in der Zwischenzeit einige Workarounds implementiert bekommen hat.

Zusammenfassung

Die Kardinalitätsabschätzungen des Optimizers sind von zentraler Bedeutung für die Effizienz der generierten Ausführungspläne. Die Statistiken sollten daher möglichst genau die Daten so repräsentieren, wie sie von der Applikation abgefragt werden, also für die Fragen, die die Applikation stellt, die richtigen Abschätzungen liefern.

Für eine effiziente und akkurate Statistikerstellung in den aktuellen Versionen von Oracle müssen bestimmte Parameter beim Aufruf von DBMS_STATS verwendet werden, das ANALYZE-Kommando sollte dafür schon lange nicht mehr zum Einsatz kommen.

Histogramme können in den Versionen vor 12c zu Schwierigkeiten bei den Abschätzungen führen, Oracle 12c führt hier einige maßgebliche Veränderungen und Verbesserungen ein.

Die korrekte Verwendung von Datentypen hilft, Probleme bei den Abschätzungen des Optimizers zu vermeiden.

Kontaktadresse:

Randolf Geist
Unabhängiger Berater
Mannheim

Telefon: +49 (0) 170-758 1171
E-Mail info@sqltools-plusplus.org
Internet: <http://oracle-randolf.blogspot.com>