

Cost Based Optimizer Grundlagen

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:

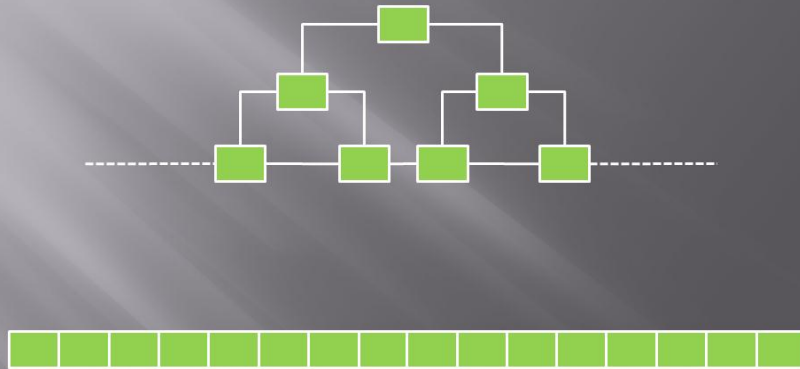
Die erste Frage nach der Anzahl der Zeilen und des Datenvolumens wird recht ausführlich behandelt, wobei vor allem die Abschätzung von Join-Operationen in Bezug auf Anzahl Zeilen und Datenvolumen auf recht groben Angaben (Statistiken) beruhen und daher potentiell ziemlich falsch sein können.



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.

Proaktives Design

Gerade bei OLTP-Anwendungen, die gezielt kleinere Datenmengen suchen, kann die Organisation von Daten einen großen Unterschied in Bezug auf die Effizienz beim Datenzugriff machen.

Daher sollten idealerweise schon während des Designs der Applikation die Daten und die wichtigsten Abfragen auf diese Daten bekannt sein.

Mit Hilfe dieses Wissens kann dann evaluiert werden, ob die Verwendung von anderen Speicherungsverfahren den Zugriff auf die Daten entscheidend verbessern kann, ohne beim Modifizieren der Daten zu viel Zeit zu kosten.

Insbesondere die Verwendung von Clustern (Index Cluster oder Hash Cluster) oder Index-Organized Tables (IOTs) bieten je nach Anwendungsfall die Möglichkeit, die Organisation der Daten proaktiv zu beeinflussen und den Zugriff auf die Daten entscheidend zu beschleunigen.

Natürlich können diese alternativen Speicherungsverfahren nicht ungesehen eingesetzt werden. Die genauen Zugriffsmuster sowohl beim Lesen als auch Schreiben der Daten müssen bekannt sein, da

ansonsten leicht das Gegenteil erreicht werden kann: Sowohl Lese- als auch Schreibzugriff können deutlich ineffizienter werden.

Weiterhin wird ein Wissen über die spezifischen Verhaltensweisen von Clustern und IOTs benötigt, um vorab abschätzen zu können, welche der alternativen Speicherungs-methoden welche Vorteile und Nachteile mit sich bringen.

Werden zum Beispiel viele sekundäre Indizes nötig sein, kann eine IOT zu insgesamt schlechterer Performance führen, da sekundäre Indizes für IOTs grundsätzlich anders funktionieren als für normale Tabellen.

Cluster können aus verschiedenen Gründen sehr ineffizient werden und erlauben bestimmte Operationen und Zugriffsmuster, je nach Variante, nicht. Darüber hinaus unterstützen derzeitige Versionen keine Partitionierung von Clustern.

Insgesamt scheint es also sinnvoll zu sein, bereits während der Designphase einer Applikationen einen Berater hinzuzuziehen, der diese Kernfragen bezüglich der Daten und ihrer Speicherung in der Datenbank klären kann und ein optimales logisches sowie physisches Design vorschlagen kann.

Kontaktadresse:

Randolf Geist
Unabhängiger Berater
Mannheim

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