

7.3.5 Vorabanalyse mit dem Code Inspector

Der Code Inspector (SCI) wurde in den vorangegangenen Kapiteln immer wieder erwähnt. Er stellt ein paar nützliche Prüfungen zur Verfügung, mit denen Sie einige Probleme schon in einem automatisierten Vortest erkennen können:

- Probleme mit der WHERE-Klausel einer SELECT-Anweisung,
- Anweisungen auf gepufferte Tabellen, die den Puffer nicht nutzen können,
- Operationen auf internen Tabellen, die nicht optimiert sind.

Es ist sinnvoll, wenn Sie vor der Analyse den Code Inspector schon auf die Bausteine anwenden, die bei dem Testfall ausgeführt werden.

7.3.6 Erwartungen und Rahmenbedingungen

Schließlich sollten Sie noch prüfen, ob es Erwartungen oder Rahmenbedingungen an die Performance der Anwendung gibt. Vielleicht steht für die Anwendung nur ein bestimmtes Zeitfenster zur Verfügung oder es gibt eine ältere Lösung der Aufgabe, die in einer bestimmten Zeit fertig wurde.

Falls es solche Erwartungen gibt, können Sie daraus Anforderungen an die Optimierung ableiten. Schon an den Gesamtlaufzeiten können Sie dann ersehen, wie realistisch diese Erwartungen sind.

Eine gute Vorbereitung ist enorm wichtig für eine gute Analyse. Falls Sie den falschen Testfall auswählen oder nicht reproduzierbare Daten analysieren, verschwenden Sie nur unnötig Zeit, ohne sinnvolle Ergebnisse zu erhalten.

7.4 Analyse anhand der SQL-Trace

Die große Bedeutung der SQL-Trace zeigt sich erst jetzt bei der Performanceanalyse. Die Informationen, die die SQL-Trace liefert, erlauben eine sehr detaillierte Bestimmung des Optimierungspotenzials.

Bisher wurde die SQL-Trace nur genutzt, um Fragen wie folgende zu beantworten:

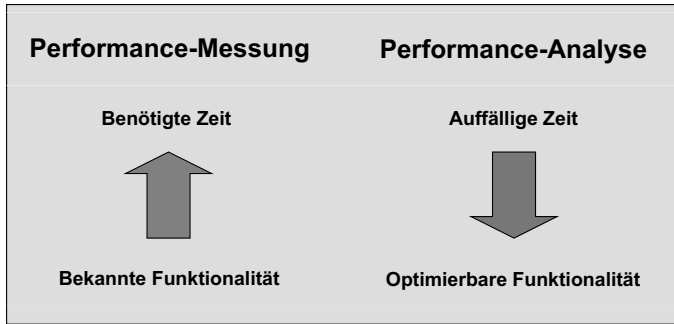
- Wie oft wird eine bestimmte Anweisung ausgeführt?
- Wie lange dauert sie?
- Wie viele Sätze liest oder ändert sie?

Nun soll die Anwendung als Ganzes analysiert werden. Die Fragen gehen vom Ganzen zum Detail:

- Wie lange dauert die gesamte Anwendung?
- Welche Anweisungen tragen besonders viel dazu bei?
- Haben diese Anweisungen Optimierungspotenzial, verstoßen sie gegen eines oder mehrere Performanceziele?
- Wenn ja, wo liegt das Problem und was ist die Ursache?

Abb. 7-1

Messung oder Analyse der Performance



Die Vorgehensweise hat sich umgekehrt. Es ist nicht mehr eine bestimmte Anweisung von vornherein interessant, sondern es sind diejenigen Anweisungen am interessantesten, die sich am stärksten verbessern lassen. Das ist die Aufgabe der Performanceanalyse.

Nach den umfangreichen Vorbereitungen in den vorhergehenden Kapiteln gilt: Ist ein Problem erst mal identifiziert, dann kann anhand der Programmierempfehlung meist relativ schnell eine Lösung gefunden werden.

Bedeutung der Performanceziele bei der SQL-Trace

Abb. 7-2

Verdichtung nach SQL-Anweisungen mit Checks

Wie lassen sich die fünf Ziele der Performanceanalyse mit der SQL-Trace überprüfen? Die geeignete Darstellung des Trace-Ergebnisses ist die Verdichtung nach SQL-Anweisungen:

Verdichtete SQL-Anweisungen - sortiert nach duration

Executions	Identical	Duration	Records	Time/exec	Rec/exec	Avg_time/r	Min_time/r	Length	Btype	Tabtype	Objektname	SQL-Statement
1	0	4.589.291	0	4.589.291	0,0	4.589.291	4.589.291	594		TRANSP	DD03L	SELECT WHERE "
1	0	922.658	64.313	922.658	64.313,0	14	14	184	gen	TRANSP	DD03L	SELECT FROM "D
3	33	543.707	7.129	181.236	2.376,3	76	62	594		TRANSP	DD03L	SELECT WHERE "
1.361	0	351.711	1.355	258	1,0	260	133	594		TRANSP	DD03L	SELECT WHERE "
200	95	55.811	290	279	1,0	279	186	594		TRANSP	DD03L	SELECT WHERE "
1	0	40.500	1.361	40.500	1.361,0	30	30	315		TRANSP	DD02L	SELECT WHERE "
1	0	18.214	1.000	18.214	1.000,0	18	18	594		TRANSP	DD03L	SELECT WHERE "
3	0	2.689	3	896	1,0	896	557	268	sng	TRANSP	TRDIR	SELECT WHERE "
2	50	2.142	200	1.071	100,0	11	11	184	gen	TRANSP	DD35L	SELECT WHERE "
2	0	737	0	369	0,0	369	330					COMMIT WORK
1	0	493	1	493	1,0	493	493	348		VIEW	TRDIR	SELECT WHERE "
1576	12	6.527.953	75.562	4.142	47,9	86						

3

5b

2

1

5a

Verstöße gegen die fünf Ziele lassen sich anhand der folgenden sechs Tests finden:

Test	Test	Beschreibung
		Sortieren Sie nach <i>Duration</i> , also Laufzeit, absteigend. Bestimmen Sie, welche Beiträge mehr als 1% der Gesamtlaufzeit ausmachen.
0	Einsparpotenzial	Überprüfen Sie alle teuren Beiträge auf ihre Notwendigkeit.
1	Min.time/r = langsame Ausführungen	Betrachten Sie die Einträge in der Spalte <i>Min.time/r</i> , Werte größer 10.000 µs sollten genauer untersucht werden. Sie weisen auf mögliche Probleme bei der Zugriffseffizienz hin.
2	Records = zu viele Sätze	Überprüfen Sie die Spalte <i>Rec</i> , d.h. Anzahl der bearbeiteten Sätze. Alle Einträge größer 50 sollten näher betrachtet werden.
3	Executions = zu viele Ausführungen	Analysieren Sie die Spalte <i>Exec</i> , d.h. die Ausführungen. Alle Zeilen mit einem Wert größer 50 oder einem noch zu bestimmenden Wert können vielleicht verbessert werden.
4	Length = Satzbreite	Große Werte von <i>Length</i> (> 500 Byte) deuten auf breite Tabellen hin, bei ihnen ist eventuell die Verwendung einer Feldliste sinnvoll.
5a	Btype = Pufferungsart oder -möglichkeit	Einträge in der Spalte <i>Btype</i> deuten auf ein Problem mit der Tabellenpufferung hin.
5b	Identical = einsparbare identische Ausführungen	Einträge in der Spalte <i>Identical</i> deuten auf Pufferungsbedarf, mit Tabellenpuffer oder Pufferbaustein, hin.

Tab. 7-4

Ziele der Performance angewandt auf SQL-Trace

- Das Ziel 0 besitzt eine Sonderrolle, denn es lässt sich nicht an einer technischen Kenngröße festmachen, sondern es müssen alle teuren Beiträge bezüglich der Notwendigkeit überdacht werden.
- Das Ziel 1 lässt sich recht gut mit der minimalen Zeit pro Satz überprüfen. Entscheidet sich der Optimierer für einen schlechten Zugriffspfad, dann steigt die Zeit pro Satz meist stark an.
- Das Ziel 2 bezieht sich auf die benötigte Menge von Sätzen. Diese lässt sich sehr gut überprüfen. Es muss lediglich der richtige Grenzwert gefunden werden, siehe unten.
- Ähnlich verhält es sich bei Ziel 3, der Zahl der Ausführungen. Auch hier muss der richtige Grenzwert festgelegt werden.
- Es genügt das vierte Ziel, die Skalierbarkeit, anhand der ABAP-Trace und des im Abschnitt 7.7 beschriebenen Tools zu testen.
- Das neue Ziel 4 bezieht sich auf die Optimierung der Satzbreite.
- Die Tests 5a und 5b beziehen sich beide auf die Pufferung. Sie können als Verstöße gegen die Ziele 0 und 1 angesehen werden. Zum

einen sind Zugriffe auf die Datenbank unnötig, wenn sie eigentlich auf den Puffer gehen sollten. Zum anderen sind sie natürlich schneller, wenn sie den Puffer nutzen. In der Reihenfolge sind sie mit Absicht hinter den Tests 2 und 3 angeordnet, da auch vom Puffer nicht unnötig viel und unnötig oft gelesen werden sollte.

Richtige Wahl der Grenzwerte

Die obigen Grenzwerte bei den Tests 1, 2, 3 und 4 sind Vorschläge, die gegebenenfalls an den Kontext angepasst werden müssen.

Der Grenzwert $10.000\mu\text{s}$ bzw. 10 ms für Min.time/r ist relativ großzügig. Jeder gute Zugriff sollte um einiges schneller sein. Auf einer aktuellen Hardware sind Zugriffszeiten pro Satz von $500\mu\text{s}$ möglich und Mengenzugriffe sind noch weit schneller. Schwierige Suchen, Plattenzugriffe und ältere Hardware liefern natürlich höhere Werte. Der Grenzwert kann deshalb nur eine Auswahlhilfe für die weitere Untersuchung sein. Er sollte nicht zu klein gewählt werden, um nicht zu viele Anweisungen testen zu müssen. Mit einem kleineren Grenzwert sind Sie sicherer, dass Ihnen kein langsamer Zugriff entgeht.

Der Grenzwert für die Anzahl der gelesenen Sätze hängt von der Aufgabe und dem zu erwartenden Datenvolumen ab. Ein Wert von 50 ist bei der transaktionalen Verarbeitung eines kleinen Objekts sinnvoll. Bei größeren Objekten oder einer Reportingaufgabe können weit größere Werte richtig sein.

Bei dem Grenzwert für die Zahl der Ausführungen verhält es sich ähnlich. Auch hier sind größere Grenzwerte bei komplexeren Aufgaben sinnvoll. Die Anzahl der Ausführungen sollte auch weniger schnell zunehmen als die Anzahl der Sätze, da eine Array-Verarbeitung genutzt werden sollte.

Falls Sie sich bei den Tests 2 und 3 für keinen Wert entscheiden können, dann überprüfen Sie einfach, welche größten Werte vorkommen, und bestimmen daraus, ob diese sinnvoll sind oder nicht.

Bei der Satzbreite können Sie auf einen Grenzwert auch ganz verzichten und einfach bei den teuersten Anwendungen nachsehen, ob eine Feldliste verwendet wird.

Zusammenhang mit den Goldenen Regeln

Offensichtlich hängen die Ziele der Performanceanalyse bei der SQL-Trace direkt mit den Goldenen Regeln der Datenbankprogrammierung zusammen. Nur das Ziel 0 entspricht keiner Regel. Die Diskussion der einzelnen Regeln in den Abschnitten 4.2 bis 4.9 dient als Grundlage für die Ursachenfindung.

Performanceziel	Test in SQL-Trace	Goldene Regel Nr.	Erklärung in Abschnitt
0	Große Zeiten		
1	Min.time/r	1	4.2
2	Records	2	4.3
3	Executions	3	4.4
4	Length = Satzbreite	4	4.5
5a	Btype	5	5.7
5b	Identical		5.7, 5.8

Tab. 7-5

Ziele der Performance und Goldene Regeln

Um die Ursachen der Probleme zu finden, sind noch weitere Untersuchungen nötig. Sie können nach Gefühl bzw. eigenen Erfahrungen vorgehen und die Zusammenfassung des vierten Kapitels als Hilfsmittel nutzen. Jedoch ist gerade die Untersuchung der Zugriffseffizienz (Ziel 1) nicht trivial. Eine Hilfestellung soll Ihnen die systematische Anleitung im nächsten Abschnitt geben.

Die Ziele der Analyse lassen sich in der SQL-Trace anhand der Verdichtung nach SQL-Anweisungen direkt in einzelnen Tests umsetzen. Für jedes Ziel gibt es eine Kenngröße, für die sich ein Grenzwert festlegen lässt. Ist der Grenzwert überschritten, sollten Sie die Ursache suchen.

7.5 Ursachensuche bei SQL-Problemen

Nach mehreren Versuchen habe ich mich entschlossen, folgende Beschreibung eher knapp zu halten und die Tabellen zur besseren Übersicht zu nutzen. Falls Sie selbst herausfinden wollen, was die Ursache der identifizierten Probleme ist, kann sie eine gute Unterstützung sein. Die Beschreibung deckt die meisten Möglichkeiten ab, deshalb sollte es sich auch für einen erfahrenen Leser lohnen, die Punkte einmal durchzugehen, um vielleicht noch unbekannte Ursachen zu finden.

7.5.1 Analyse der langsamen SELECTs

Die Analyse einer langsamen Ausführung ist die umfangreichste der Analysen, denn es sind mehrere Schritte nötig. Insbesondere sind die Schritte etwas schwer zu erklären; wenn Sie sie erst einmal durchgespielt haben, sind die Schritte relativ klar. Folgendes ist hier durchzuführen: