

Troubleshooting Oracle Parallel Execution Performance

Randolf Geist
Unabhängiger Berater
Mannheim, Deutschland

Schlüsselworte

Parallele SQL-Ausführung, Parallel Execution, Oracle Enterprise Edition, Analyse, Performance Troubleshooting

Zusammenfassung

Oracle Database Enterprise Edition erlaubt mittels dem Parallel Execution Feature die automatische Verteilung einer SQL-Ausführung auf mehrere Parallel Execution Server. Dabei können spezielle Performance-Probleme auftreten, die es so mit normaler serieller Ausführung nicht gibt.

In diesem Vortrag wird eine Übersicht geboten, welche Herausforderungen es mit paralleler Ausführung im Allgemeinen geben kann, und wie diese auf Oracle Parallel Execution zutreffen.

Desweiteren wird erläutert, wie diese speziellen Probleme identifiziert und analysiert werden können. Dieses Verständnis ist wichtig für eine systematische Herangehensweise beim Troubleshooting von Parallel Execution und daher empfehlenswert für alle DBAs, Entwickler und Anwender, die Parallel Execution ernsthaft verwenden wollen.

Einleitung

Parallel Execution sollte nur in speziellen Fällen eingesetzt werden, in denen vorher evaluiert wurde, dass es keine bessere Lösung gibt.

Der wichtigste Punkt ist, dass Parallel Execution im Grunde ein „Brute Force“-Ansatz ist.

Dies bedeutet, dass man nicht versucht, ein Performance-Problem durch Effizienzerhöhung zu lösen, sondern durch die Verwendung von mehr Ressourcen, insbesondere CPU und Arbeitsspeicher.

Es sollte also klar sein, dass Parallel Execution tatsächlich dazu führt, dass man insgesamt mehr und härter arbeitet im Vergleich zu einer optimalen seriellen Ausführung, vor allem durch den Overhead, der durch Parallel Execution verursacht wird.

In vielen Fällen kann man aber eine geeignete Performance-Steigerung dadurch erreichen, weniger bzw. effizienter zu arbeiten – angefangen bei effizienteren Ausführungsplänen, bis hin zu Verwendung von optimierten Speicherstrukturen in der Datenbank, wie cleveres Index-Design, Cluster oder Index Organized Tables.

Diese Optionen sollten also evaluiert werden, bevor man versucht, eine Performance-Steigerung mittels Parallel Execution zu erreichen.

Spezielle Herausforderungen

Es gibt einige Probleme, die nur auf Parallel Execution zutreffen. Die wichtigsten haben damit zu tun, wie die zu verrichtende Arbeit auf die N Prozesse verteilt wird, die gleichzeitig arbeiten sollen, um die Gesamtlaufzeit zu verringern.

Dabei gibt es zwei Grundformen:


1. Es kann passieren, dass sich ein Problem grundsätzlich nicht passend aufteilen lässt („Parallel Unfriendly“), wobei es hier noch die Variante geben mag, dass theoretisch eine Aufteilung möglich wäre, aber eine entsprechende Implementierung seitens Oracle (noch) nicht umgesetzt wurde.

Ein gutes Beispiel für diese Form ist eine Abfrage nach dem Median über einen gesamten Datenbestand. Oracle kann zwar die Datenermittlung parallelisieren (Parallel Full Table Scan), aber die eigentliche Berechnung des Median erfolgt im zentralen Prozess (Query Coordinator), der sonst nur für die Steuerung der Parallel Worker und Kommunikation mit dem Client zuständig ist. Insofern reduziert diese Parallel Execution sehr wahrscheinlich die Gesamtausführungszeit nur im geringen Maße, falls überhaupt, und wird vor allem nicht schneller, wenn man versucht, die Parallelität zu erhöhen. Der Grund hierfür liegt in diesem Fall darin, dass der gesamte Datenbestand im zentralen Prozess sortiert wird.

PARALLEL UNFRIENDLY

```
select median(id) from t2;
```

Id	Operation	Name	TQ	IN-OUT	PQ	Distrib
0	SELECT STATEMENT					
1	SORT GROUP BY					
2	PX COORDINATOR					
3	PX SEND QC (RANDOM)	:TQ10000	Q1,00	P->S	QC	(RAND)
4	PX BLOCK ITERATOR		Q1,00	PCWC		
5	TABLE ACCESS FULL	T2	Q1,00	PCWP		



2. Ein Problem kann grundsätzlich aufgeteilt werden und wird so auch von Oracle unterstützt, fällt also nicht unter 1., aber die Verteilung der Arbeit auf die N Prozesse ist nicht gleichförmig („Parallel Execution Skew“).

In diesem Fall sind nicht alle Prozesse permanent ausgelastet, so dass es zu signifikanter „Idle Time“ dieser Prozesse kommen kann, was natürlich dazu führt, dass die gewünschte Reduktion der Ausführungszeit nicht in dem Maße erreicht werden kann, wie erwartet.

Im schlechtesten Fall kann Parallel Execution in so einer Konstellation sogar langsamer als

eine serielle Ausführung werden, da Parallel Execution eben mit einigem Overhead kommt, so dass ein vergleichbarer Ausführungsplan langsamer bei Verwendung von Parallel Execution sein kann.

Parallel Execution Skew

Die unter 2. erwähnte Problematik dient besonderer Erwähnung, da sie in der Praxis mit am häufigsten auftritt und dazu führen kann, dass Parallel Execution wesentlich weniger die Ausführungszeit verringert, als theoretisch möglich wäre.

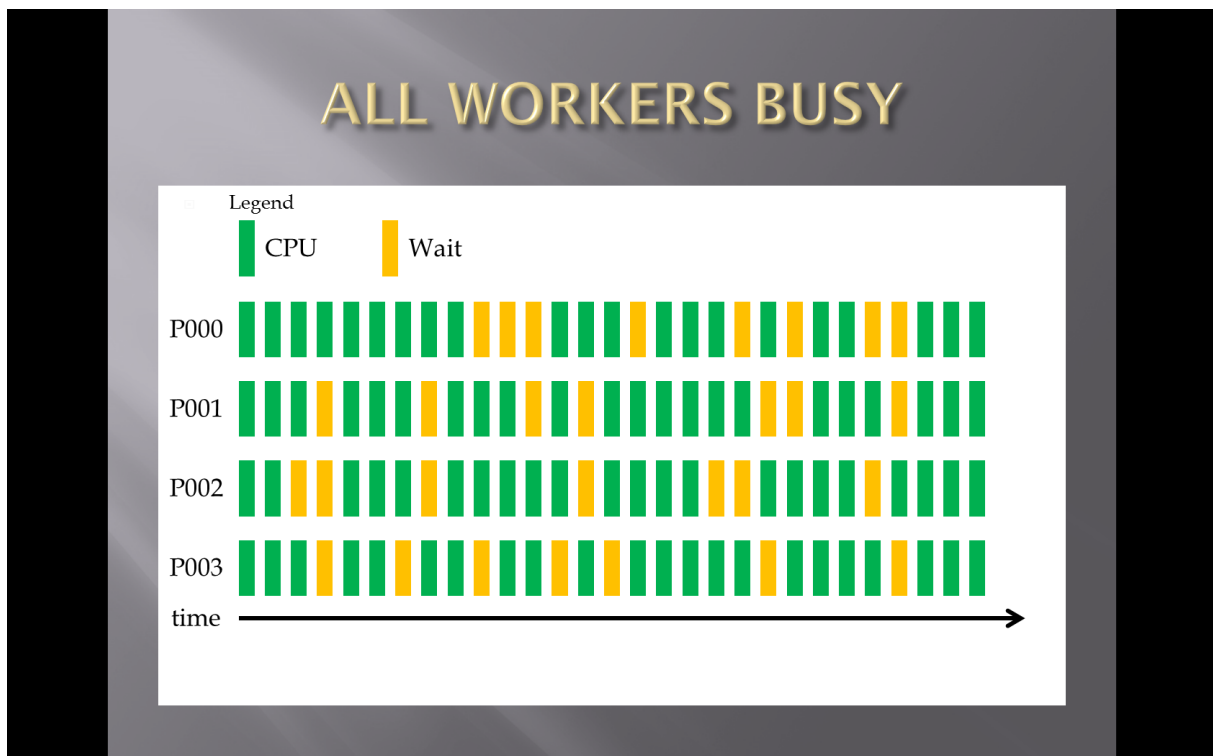
Bei der Ungleichverteilung der Arbeit auf die Prozesse gibt es wiederum zwei Arten, die man grundsätzlich unterscheiden kann.

Dazu ein paar Grafiken, die die verschiedenen Formen veranschaulichen sollen.

Gute Auslastung

Betrachtet man die Auslastung der beteiligten Prozesse, sollte sich idealerweise ein Bild ähnlich dem folgenden ergeben. Exemplarisch sind hier vier Prozesse visualisiert (P000 – P003) und die Boxen stehen für einen bestimmten Zeitraum (zum Beispiel ein Sampling der Aktivität jede Sekunde → Active Session History).

Die Farbe der Box zeigt zusätzlich an, ob der Prozess grundsätzlich auf CPU ist und arbeitet, oder auf etwas aktiv wartet zur Weiterverarbeitung – bei einer Datenbank ist dies häufig Warten auf Daten (I/O waits).



Hier sind die vier untersuchten Prozesse also grundsätzlich immer ausgelastet, und idealerweise auch meistens auf CPU.

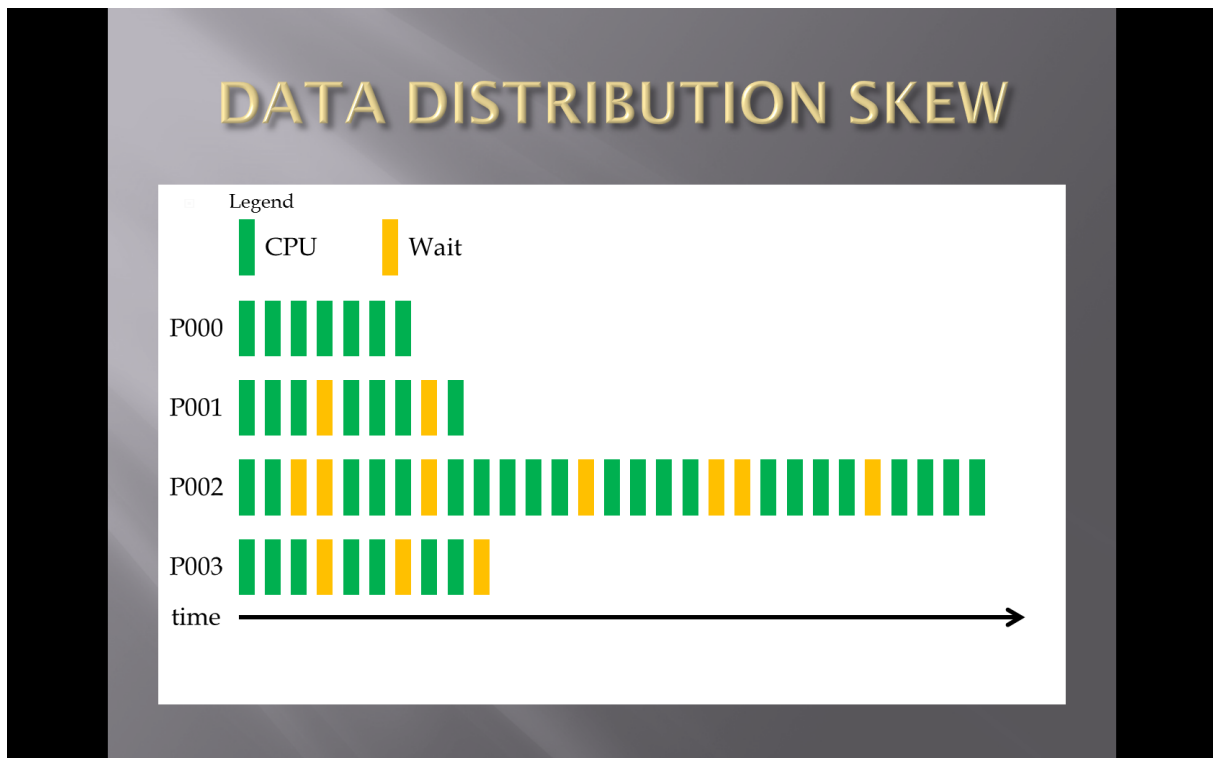
Sollten die Prozesse zwar ausgelastet sein, aber die meiste Zeit warten, dann ist eine Performance-Steigerung gegenüber serieller Ausführung auch nur sehr begrenzt möglich.

Es ist sehr wichtig zu verstehen, dass wir hier nur die Auslastung der Prozesse betrachten. Dies ist keine Bewertung der tatsächlichen Arbeit, die verrichtet wird. Insofern ist es möglich, einen völlig ineffizienten Ausführungsplan mittels Parallel Execution auszuführen, der zwar eine wunderbare Auslastung der beteiligten Prozesse erreicht, aber trotzdem viel zu lange benötigt.

Umgekehrt ist es möglich, einen ansonsten effizienten Ausführungsplan zu haben, der mittels Parallel Execution ausgeführt wird – werden dabei aber die Prozesse nicht richtig ausgelastet, ist eine Reduzierung der Ausführungszeit trotz ansonsten effizientem Ausführungsplan mittels Parallel Execution nur sehr begrenzt möglich, falls überhaupt.

Data Distribution Skew

Werden die zu verarbeitenden Daten ungleich auf die beteiligten Prozessen verteilt, dann kann folgendes Bild entstehen:

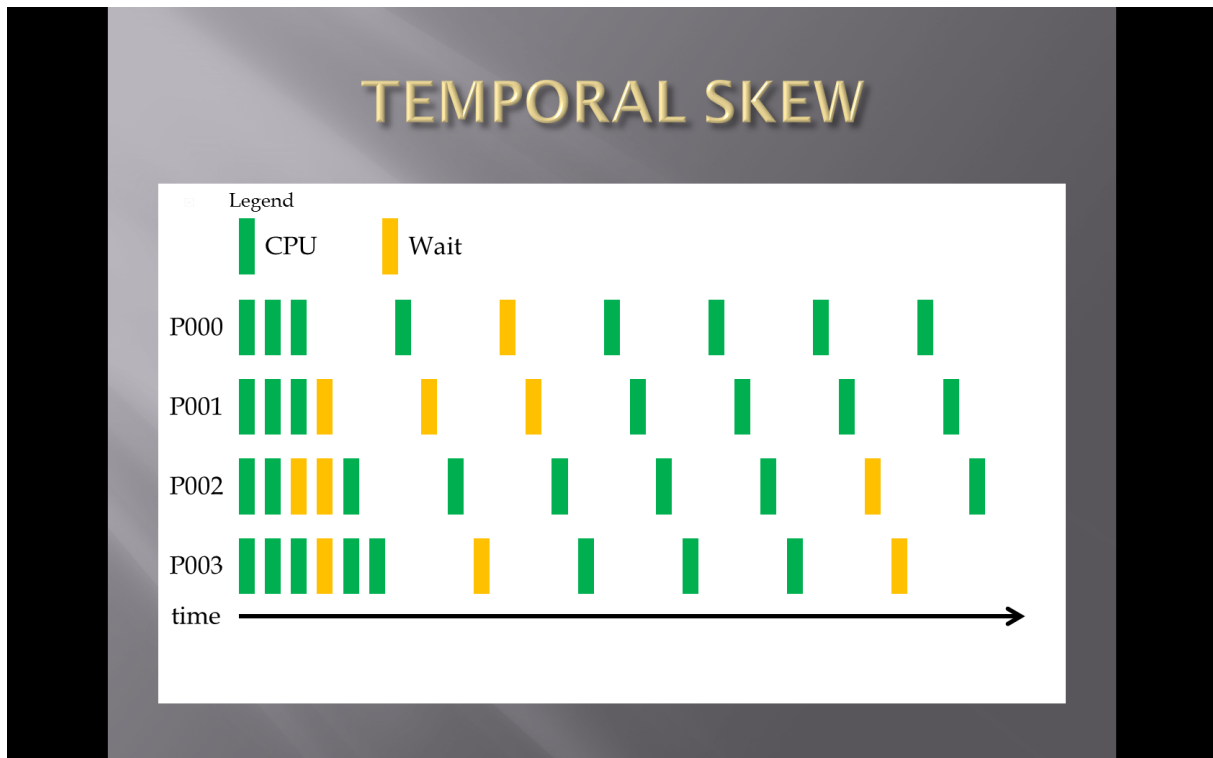


Hier ergibt sich also eine signifikante „Idle Time“ bei dreien der beteiligten vier Prozesse – und der Löwenanteil der Arbeit muss von einem der vier verrichtet werden.

Offensichtlich wird hier die Gesamtlaufrzeit von den Prozessen bestimmt, die den größten Teil der Arbeit verrichten müssen, und diese kann im schlechtesten Fall sogar gleich oder über der einer seriellen Ausführung liegen.

Temporal Skew

Es gibt noch eine andere Möglichkeit, wie die Gesamtauslastung der beteiligten Prozesse weniger als theoretisch möglich sein kann: Die zu verarbeitenden Daten werden zwar gleich verteilt, aber die Prozesse sind zwischendurch immer wieder „idle“, so dass sie zwar alle ungefähr zur gleichen Zeit fertig sind, und auch eine ähnliche Datenmenge zu verarbeiten haben, aber die Reduzierung der Gesamtlaufzeit durch die signifikante „Idle Time“ limitiert ist:



Man kann darüber diskutieren, ob es sich in diesem Fall tatsächlich um „Idle Time“ handelt, da meistens auf zu empfangende Daten von anderen Prozessen gewartet wird, insofern könnte man es auch als eine bestimmte Form des „aktiven“ Wartens bezeichnen.

Diese Form tritt übrigens häufig im Zusammenhang mit „Data Distribution Skew“ auf, falls eine Umverteilung der Daten notwendig ist (benötigt den Austausch der Daten zwischen den Prozessen), und im vorherigen Schritt ein Großteil der Daten von nur einem oder wenigen Prozessen verarbeitet wurde.

Analyse und Diagnose

Wie können nun solche Fälle diagnostiziert werden, bei denen die Auslastung der an der Parallel Execution beteiligten Prozesse nicht dem Ideal entspricht?

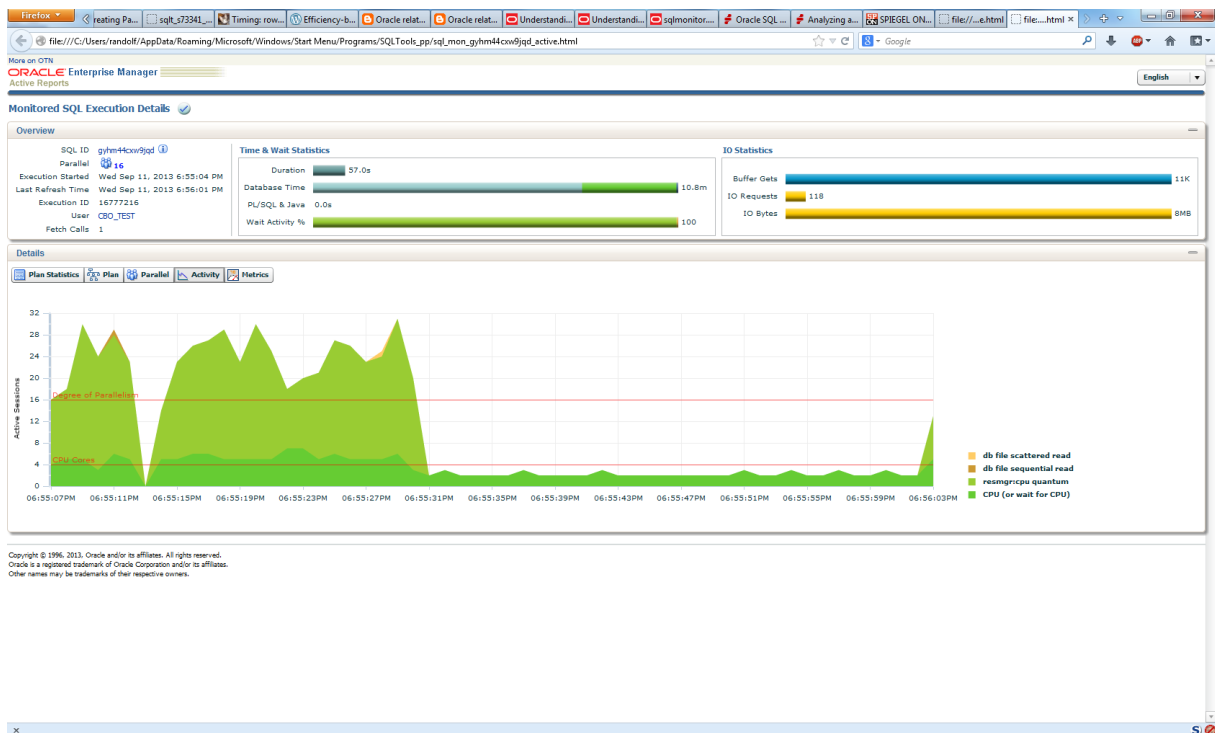
Diese Möglichkeit der Analyse ist wichtig, um zu verstehen, warum eine Parallel Execution nicht zu der gewünschten Reduzierung der Laufzeit führt.

Es muss unterschieden werden zwischen dem Fall, bei dem eine grundsätzliche Ineffizienz vorliegt – dieser zeichnet sich dadurch aus, dass „zu viel“ gearbeitet wird, und dem hier beschriebenen Phänomen, bei dem grundsätzlich „zu wenig“ gearbeitet wird.

In der Vergangenheit war mit den von Oracle zur Verfügung gestellten Mitteln eine solche Diagnose schwierig. Seit der Version 11.2 stellt das sogenannte „Real-Time SQL Monitoring“ Informationen zur Verfügung, die eine solche Analyse vereinfachen. Leider benötigt das Feature die Diagnostic + Tuning Pack Zusatzlizenz, steht also ohne diese nicht zur Verfügung.

In der sogenannten „Enterprise Manager“ oder auch „Active“-Version (und nur dort) eines „Real-Time SQL Monitoring“-Berichts gibt es den Abschnitt / Tab Aktivität (Activity).

Dieser erlaubt eine visuelle Darstellung der Auslastung der beteiligten Prozesse, und somit eine zuverlässige Aussage (mit Ausnahme von bestimmten Fällen, die ich in meinem „Parallel Execution Internals Masterclass“ Seminar erkläre), ob es signifikante Ausreißer in der Auslastung gab:



Im Bild oben kann man erkennen, dass ein Parallel Degree (DOP) von 16 angegeben ist, aber die Aktivität nur für eine bestimmte Zeit am Anfang der Ausführung im (oder sogar über) dem erwarteten Bereich lag, und für den Rest der Zeit in einem viel niedrigeren Bereich.

Dies ist die einfachste Art und Weise festzustellen, ob es ein Problem mit der Auslastung der Prozesse gab oder nicht.

Leider benötigt sie die entsprechenden Lizenzen, und ist daher nicht ganz kostengünstig.

Es gibt noch weitere Möglichkeiten der Diagnose, diese zu beschreiben, sprengt aber den Rahmen dieses Vortrags.

Problembhebung

Falls ein Problem mit der Aktivität der beteiligten Prozesse als Grund für eine zu lange Laufzeit bei Verwendung von Parallel Execution festgestellt wurde, wie kann dieses adressiert werden?

Je nach genauen Umständen gibt es unterschiedliche Lösungsmöglichkeiten, die auch mit unterschiedlichem Umsetzungsaufwand einherkommen.

Die einfachste Lösung kann in der Änderung der sogenannten Distributionsmethode bestehen (Parallel Distribution Method) – entweder durch Beeinflussung der Datenvolumenabschätzungen des Optimizers, oder explizite Angabe des PQ_DISTRIBUTE-Hints.

Jedoch ist diese Maßnahme nicht in allen Fällen sinnvoll und kann auch zu extremer CPU- und Speicher bzw. TEMP-Auslastung führen.

Andere Maßnahmen können die Limitierung des Skews durch Beeinflussung der Join-Reihenfolge sein, oder im aufwendigsten Falle das spezielle Umschreiben der Abfrage, um das Problem zu begrenzen oder sogar zu umgehen, je nach genauer Sachlage.

Möglicherweise kann das Problem auch nur umgangen werden, indem die Art und Weise, wie die Daten in der Datenbank repräsentiert werden, angepasst wird.

Kontaktadresse:

Randolf Geist
Unabhängiger Berater
Mannheim

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