

Möglichkeiten zur Prozesskontrolle in Datenbank-Batchverarbeitungen

Dr. Kurt Franke
Cellent Finance Solutions GmbH
Stuttgart

Schlüsselworte

Jobsteuerung, Prozesskontrolle, Hintergrundverarbeitung, asynchron

Einleitung

Datenbank-Batchverarbeitungen laufen unabhängig von aktuell bestehenden Verbindungen in der Datenbank im Hintergrund ab. Es ist sogar so, dass die Prozesskontrolle selbst in irgendeiner Form im Hintergrund läuft, wobei die von ihr asynchron angestarteten Batchjobs auch im Bezug auf die Prozesskontrolle im Hintergrund laufen. Zur Erzeugung der Sessions / Prozesse für die Batchjobs bietet sich in heutigen Zeiten die Verwendung von `dbms_scheduler` mit seinen vielfältigen Möglichkeiten an. Das ist jedoch nicht zwingend – jede Methodik, die das asynchrone Starten einer Session mit Vorgabe der darin auszuführenden Verarbeitung erlaubt, ist dafür geeignet.

Solche komplett innerhalb einer Datenbank laufenden Batchjobs benötigen eine Möglichkeit, mit Fehler abgebrochene von noch laufenden Batchverarbeitungen zu unterscheiden. Erfolgreich beendete Verarbeitungen werden durch einen bei Beginn zugeordneten und am Ende abgeschlossenen Eintrag in einer Logtabelle erkannt, ebenso Verarbeitungen, die mit einem normalen Fehler abgebrochen sind und ihren Eintrag in der Logtabelle noch abschließen konnten – hier natürlich mit einem Fehlerstatus. Bei Auftreten von Oracle-Exceptions, die die zugehörige Datenbank-Session terminieren (z. B. ORA 04030, Ora 07445), ist dies jedoch nicht mehr möglich. Für solche Fälle wird ein Mechanismus zur Erkennung benötigt, ob für als laufend gekennzeichnete Verarbeitungseinträge die zugehörigen Prozesse/Sessions noch existieren. Im folgenden werden 2 verschiedene Möglichkeiten für ein Handling zur Prozesskontrolle von Batchjobs beschrieben.

Die erste Variante verwendet die Views `V$SESSION` / `V$PROCESS` bzw. Sub-Views darauf, die auf die zur gehandelten Batch-Applikation gehörenden Sessions / Prozesse limitiert sind. In der Statustabelle für die Batchverarbeitungen sind geeignete Columns integriert, um direkte Joins mit diesen V\$-Views zu ermöglichen und somit eine Verknüpfung der Session-Information mit der Verarbeitungsinformation bereitzustellen.

Die zweite Variante verwendet einen ganz anderen Ansatz, der davon ausgeht, dass ein Zugriff auf die Views `V$SESSION` / `V$PROCESS` bzw. Sub-Views darauf nicht zur Verfügung steht. Man braucht dazu einen Object-Typ, dessen Eindeutigkeit bei der Erzeugung sichergestellt ist und der die Lebensdauer einer Session hat und bei Session-Beendigung automatisch bereinigt wird. Einen solchen Object-Typ bietet die Verwendung von Locks, die über das `dbms_lock`-Package mit `Duration = SESSION` erzeugt werden. Zur sicheren Zuordnung eines Locks zu einer Verarbeitung muss für jede Verarbeitung inclusive ihren klassifizierenden Parametern ein eindeutiger Lockname definiert werden, wodurch dann die Existenz/Nichtexistenz eines solchen Locks äquivalent zur Existenz/Nichtexistenz einer laufenden Verarbeitung ist.

Verwendung von V\$SESSION etc.

Die Verwendung von einigen V\$-Views ermöglicht eine umfassende Überwachung der Sessions / Prozesse einer Datenbank-Application. Es können dabei grundsätzlich alle existierenden Sessions einer Datenbank-Batch-Verarbeitung identifiziert werden.

Voraussetzung ist hier, dass Zugriffsrechte auf die benötigten V\$-Views erteilt werden.

Wegen der Wiederverwendung von Session-ID's (sid) durch den Datenbank-Kernel ist diese für sich alleine jedoch nicht als eindeutige Erkennungsgröße geeignet. Die serial#, die oft zusammen mit der sid benötigt wird, muss im Verlauf einer Session nicht konstant sein, sondern kann sich unter bestimmten Bedingungen ändern. Damit ist sie zusammen mit der sid als Erkennungsgröße nicht geeignet. Es gibt aber Werte, die sich im Laufe einer Session nicht ändern und die deshalb dafür genutzt werden können. Dazu gehören die logon_time aus v\$session und die spid aus v\$process.

Bei Datenbanken, die unter Unix-System betrieben werden, ist in der spid die Operatingssystem-Process-Id des Serverprocesses der Session hinterlegt, beim Betrieb unter einem Windows-System wird dort die Id des Threads des einzigen Datenbank-Processes einer Instanz hinterlegt, der die Session serverseitig abbildet.

Es werden also die Views V\$SESSION und V\$PROCESS benötigt, die über v\$process.addr = v\$session.paddr gejoined werden. Zur Einschränkung auf die jeweils eigene Session wird einfach noch V\$MYSTAT über v\$session.sid = v\$mystat.sid AND v\$mystat.rownum = 1 hinzugejoined.

Für eine verbesserte Übersicht kann auch noch das Package dbms_application_info zum Setzen des module-Wertes in V\$SESSION eingesetzt werden. Als module-Wert wird ein die Application und die gerade laufende Verarbeitung bezeichnender Text verwendet. Der module-Wert kann dann als weitere Einschränkung bei der Auflistung der zur Application gehörigen Sessions / Prozesse verwendet werden.

Wenn eine existierende interactive Session mehrfach hintereinander für verschiedene Verarbeitungen genutzt wird, wie es insbesondere bei Tests der Prozesskontrolle vorkommt, um den betreffenden Ablauf im Vordergrund zu haben, muss sichergestellt sein, dass ein solcher module-Wert am Ende einer Verarbeitung wieder auf den Vorgängerwert zurückgesetzt wird, um Probleme bei der Erkennung der zur Application gehörigen Sessions / Prozesse zu vermeiden. Es muss ein Exceptionhandler verwendet werden, der sicherstellt, dass das Wiederherstellen des module-Wertes auch im Fehlerfalle erfolgt.

```
BEGIN
  dbms_application_info.read_module(saved_module_name,saved_action_name);
  appl_info_read := TRUE;
  dbms_application_info.set_module('BATCH: ' || jobnam, '');
  -- hier findet die Verarbeitung statt
  IF appl_info_read THEN
    dbms_application_info.set_module(saved_module_name);
  END IF;
EXCEPTION WHEN OTHERS THEN
  IF appl_info_read THEN
    dbms_application_info.set_module(saved_module_name);
  END IF;
END;
```

Insbesondere bietet die Verwendung dieses Verfahrens auch die Möglichkeit, eigene Views zu generieren, die Joins von V\$SESSION und V\$PROCESS mit applicationseigenen Status-Tabellen durchführen, um so auch eine applications-spezifische Sicht auf die verarbeitenden Sessions / Prozesse bereitzustellen. Voraussetzung dazu ist, dass die für den Join benötigten Informationen (sid etc.) auch in den jeweiligen Status-Tabellen hinterlegt werden. Dies eröffnet den DBA's und auch den Applikations-Administratoren besonders einfache Möglichkeiten zum Erhalt notwendiger Informationen.

Grundsätzlich kann festgestellt werden, dass dieses Verfahren komplett ohne den Einsatz von Verarbeitungslocks für die Prozesskontrolle auskommt.

Zur Sicherstellung, dass eine bestimmte Verarbeitung nicht gleichzeitig mehrfach läuft, muss ein zusätzlicher Mechanismus verwendet werden. Dies kann z. B. folgendermaßen durchgeführt werden: Vor Beginn der Überprüfung wird ein Exklusivlock auf die Statustabelle der Jobs angefordert. Nach Erhalt wird geprüft, ob es für die aktuell zu startende Verarbeitung bereits einen aktiven Batchjob gibt – wenn nicht, wird einer erzeugt. Nun wird der Exklusivlock wieder freigegeben.

Eingrenzung der Views auf Applications-Prozesse

Wenn eine Application nicht Zugriff auf Informationen über alle Datenbank-Sessions erhalten soll, bietet sich die Möglichkeit, in einem privilegierten Schema – z. B. SYSTEM – eingeschränkte Views anzulegen, in denen nur Sessions / Prozesse zu finden sind, die zu einer Applikation gehören. Dies kann beispielsweise über eine Einschränkung des Session-Usernames erfolgen, von dem man üblicherweise diejenigen kennt, die zu einer Datenbank-Application gehören. In diesem Falle werden Zugriffsrechte an die Applications-Schemata nur für diese eingeschränkten Views erteilt.

Durch die Verwendung solcher eingeschränkter Views wird die Methodik der Prozesskontrolle nicht verändert, sondern nur die Information selbst auf erlaubte Prozess limitiert. Ein derartiges Verfahren kann grundsätzlich für mehrere verschiedene Datenbank-Batch-Applikationen in einer einzigen Datenbank erfolgen.

Verwendung von Locks mit Session-Duration und wohldefinierten Namen

Ein ganz anderer Ansatz ist die Verwendung von Locks mit der Lebensdauer SESSION, die über einen wohldefinierten Namen einer bestimmten Verarbeitung zugeordnet sind.

Dafür wird nur das Ausführungsrecht auf das Package DBMS_LOCK benötigt, um solche Locks zu erzeugen und abfragen zu können.

Dieses Verfahren eröffnet jedoch grundsätzlich keine direkte Möglichkeit, die zu einer Datenbank-Application gehörigen Sessions / Prozesse aufzulisten. Es ist ohne zusätzlich Rechte nur möglich, die Existenz eines solchen Locks zu prüfen und daraus auf die Existenz einer aktiven Verarbeitung zu schließen. Erst mit Zugriffs-Rechten auf DBA_LOCK und DBMS_LOCK_ALLOCATED kann die SID der betreffenden Session herausgefunden werden und dann weiter über V\$SESSION und V\$PROCESS zusätzliche Information, wobei man bei Erteilung von solchen Rechten dann auch alternativ die Rechte auf einige V\$-View erteilen kann und mit dem oben beschriebenen Verfahren arbeiten.

Was dieses Verfahren jedoch bereitstellt, ist eine Möglichkeit zur Erkennung, ob eine Verarbeitung gerade aktiv ist, erfolgreich beendet wurde, mit Fehler abgebrochen ist, oder gar die zugehörige Verarbeitungs-Session überhaupt nicht mehr existiert (nach ORA-00600 etc.). Damit können

gleichzeitige Mehrfachläufe sicher verhindert werden. Über geeignete Abhängigkeits-Definitionen kann auch verhindert werden, dass gleichzeitig verschiedenartige Verarbeitungen laufen, die die gleichen Daten manipulieren. Somit kann grundsätzlich ein korrekter automatisierter Ablauf sichergestellt werden.

Programmatisch muss dazu sichergestellt werden, dass eine Verarbeitung nur dann startet, wenn sie zuvor den zugehörigen Lock (mit dem zugehörigen Locknamen) erhalten hat. Als abschließende Aktion gibt eine Verarbeitung den allokierten Lock frei, nachdem sie die applicatonseigene Statustabelle aktualisiert hat. Ein zeitgesteuerter kurzperiodischer Datenbank-Job prüft in der Statustabelle als gerade laufend gekennzeichnete Jobs auf die Existenz des zugehörigen Locks – wenn eine solcher nicht existiert, wird der betreffende Job-Eintrag in der Statustabelle auf einen speziellen Abbruch-Status aktualisiert, der als Kennzeichnung dafür dient, dass die zugehörige Session ohne eigene Abschlussmeldung verschwunden ist.

Durch die zeitnahe Aktualisierung der Jobstati in der Statustabelle kann diese Statustabelle selbst mit allen als RUNNING gekennzeichneten Einträgen als Liste der aktiven Verarbeitungen betrachtet werden und somit als Analogon einer Session/Prozess-Liste dienen. Man muss sich jedoch dabei stets bewußt sein, dass diese Liste gerade bezüglich verschundener Sessions / Prozesse nicht unbedingt aktuell ist. Insbesondere kann so der Eindruck entstehen, dass manche Verarbeitungen in einem Überlappungsbereich gleichzeitig laufen, obwohl eine der Verarbeitungen einen Abbruch mit Verschwinden der Session hatte und der Status nur noch nicht aktualisiert wurde. Vermeiden kann man dies, wenn jede zu startende Verarbeitung nach Erhalt des Verarbeitungslocks zunächst einen Exklusiv-Tablelock auf die Statustabelle anfordert, nach dessen Erhalt zunächst alle als RUNNING markierten Jobs auf darauf überprüft, ob ein Lock und damit eine zugehörige Session existiert und bei Nichtvorhandensein den speziellen Abbruch-Status für verschwundene Sessions setzt, und erst danach den eigenen Statureintrag auf RUNNING setzt und mit der Verarbeitung beginnt.

Das alleinige Verwenden von Verarbeitungslocks bietet jedoch noch keine Möglichkeit, den Einträgen in der Statustabelle die Sessions / Prozesse der Datenbank zuzuordnen. Auch wenn diese Möglichkeit für die Applikation selbst wegen nicht ausreichenden Rechten nicht besteht, kann über eine solche Möglichkeit dem DBA doch die Arbeit erleichtert werden, wenn z. B. eine bestimmte Verarbeitung abgebrochen werden muss.

Eine solche Erleichterung bietet z. B. eine SID-Column in der Statustabelle, die beim Erzeugen eines neuen Eintrags über SYS_CONTEXT('USERENV','SID') befüllt wird. Man muss sich hier jedoch bewußt sein, dass diese SID nur für Verarbeitungen im Status RUNNING gültig ist, da die SID's von Oracle immer für neue Sessions wiederverwendet werden. Auch die Verwendung von dbms_application_info.set_module() erleichtert die Zuordnung – allein oder zusammen mit einer SID in der Statustabelle.

Nicht zuletzt erlaubt auch die Abfrage der existierenden Verarbeitungslocks die Bestimmung der SID einer Verarbeitung sofern ausreichende Privilegien zur Verfügung stehen. Dies ist über ein Statement wie das folgende möglich:

```
SELECT la.name,  
       l.session_id AS hold_by_session_id,  
       l.blocking_others,  
       la.expiration  
FROM dbms_lock_allocated la,  
     dba_lock l  
WHERE la.expiration > sysdate  
      AND la.name LIKE 'PROCESS?_CONTROL.?' ESCAPE '?'
```

```
AND l.lock_type = 'PL/SQL User Lock'  
AND l.mode_held = 'Exclusive'  
AND ltrim(rtrim(to_char(la.lockid,'999999999999'))) = l.lock_id1  
;
```

Eine Einschränkung auf Exclusic-Locks darf natürlich nur vorgenommen werden, wenn der Verarbeitungs-Lock-Mechanismus nur solche verwendet, wie es bei einer einfachen Prozesskontrolle nur für Batchverarbeitungen ohne integriertes Handling für GUI-Zugriffssperren immer sinnvoll erscheint weil sich Batchverarbeitungen immer hinreichend qualifizieren lassen, um eindeutig zu sein, während dies bei gleichartigen vielfachen GUI-Änderungs-Zugriffen nicht einfach möglich ist. Deshalb müssen in solchen Fällen auch die anderen möglichen Lockmodi mit betrachtet werden. Da solche Locks nur existieren, solange die erzeugende Session existiert, kann ein Join mit V\$SESSION allein mit der SID als Joinbedingung erfolgen und liefert damit dem DBA eine Verknüpfung von Session-Attributen mit den Verarbeitungen.

Zusammenfassung

Es wurden 2 unterschiedlich ansetzende Verfahren zur Prozesskontrolle vorgestellt. Beide ermöglichen einer Datenbank-Batch-Applikation die Überwachung und Steuerung von durchzuführenden Verarbeitungen, die von einer zentralen Steuer-Session bei Bedarf angestarten werden, wobei eine solche zentralen Steuer-Session nicht kontinuierlich existieren muss, weil die Steuer-Information jederzeit abgefragt werden kann.

Die Variante mit directem Zugriff auf V\$SESSION / V\$PROCESS bzw. auf Sub-Views darauf mit Einschränkung auf Applikations-Sessions/Prozesse erlaubt auch dem Applikations-Administrator ohne DBA-Rechte Zugriff auf Session-Attribute der aktiven Batch-Verarbeitungen. Auch für die Implementierung der Steuerung ergeben sich einfachere und mehr Möglichkeiten. Sie setzt jedoch die Erteilung von Zugriffsrechten auf diese V\$-Views entweder an das/die Applikations-Schemata oder an ein zetrnales Schema zur Erstellung von EinschränkungsvIEWS voraus.

Die Variante mit Einsatz von Verarbeitungslocks mit Session-Duration und wohldefinierten Namen benötigt wesentlich geringere Rechte – Ausführungsrechte aus das Package DBMS_LOCK sind hier ausreichend. Dafür werden DBA-Rechte benötigt, um die Sessions mit ihren Attributen zu identifizieren, welche gerade aktive Verarbeitungen durchführen. Die Implementierung der Prozesskontrolle und Steuerung ist in dieser Variante etwas anspruchsvoller.

Kontaktadresse:

Dr. Kurt Franke
Cellent Finance Solutions GmbH
Calwer Straße 33
D-70173 Stuttgart

Telefon: +49 (0) 711-222992-676
Fax: +49 (0) 711-222992-899
E-Mail Kurt.Franke@cellent-fs.de
Internet: www.cellent-fs.de