Vom Suchen und Finden – Oracle Traces auf der Spur

Katja Werner
Muniqsoft GmbH
Unterhaching

Schlüsselworte

Trace, Dump, Event, Oradebug

Einleitung

Traces – auch die von Oracle Datenbanken - sind eine Geschichte für sich. Sie stecken voller Informationen und berühren so unterschiedliche Bereiche, dass es einerseits abschreckend und andererseits äußerst faszinierend ist, sich näher mit ihnen zu beschäftigen. Oft ist Datenbankadminstratoren nur bekannt, dass SQL-Net-Verbindungen oder SQL Statements sowie Sessions getract werden können. Jedoch: Oracle Traces können auf einem weitaus größeren Feld eingesetzt werden als gemeinhin bekannt und es ist sehr spannend, dieser Datenflut Herr zu werden und sie zu entdecken. Neben einer schnelleren Problemlösung kommt der Forscher durch Traces dem auf die Spur, „was die Datenbank im Innersten zusammenhält". Von dieser Reise handelt der Vortrag „Vom Suchen und Finden – Oracle Traces auf der Spur".

To Trace or Not to Trace

Traces sind sinnvoll, aber es nicht sinnvoll, sie in jeder Lebenslage einzusetzen. Traces eignen sich sehr gut, wenn die Problemstellung bereits konkret formuliert wurden. Zum Beispiel für die Frage: Warum nimmt der Optimizer diesen Execution Plan und nicht den anderen? Dahingegen ist ein Trace für die generelle Frage, warum die Datenbank langsam ist, ungeeignet, denn die Informationsflut ist in diesem Fall unüberschaubar.

Oracle Traces – was hätten Sie den gern?

Bezüglich Oracle Traces kann unterschieden werden in Traces, die in Folge von Oracle Diagnostic Events erzeugt werden und in Traces, die in der jeweiligen Komponente bzw. im entsprechenden Utility mit einem Parameter „angeschaltet" werden. Der Output ist in beiden Fällen ähnlich: ein Trace- oder Dumpfile mit vielen Informationen.

Wenn man danach sucht, findet man an allen Ecken und Enden der Oracle-Datenbank Traces und Dumps wie Abbildung 1 veranschaulicht.

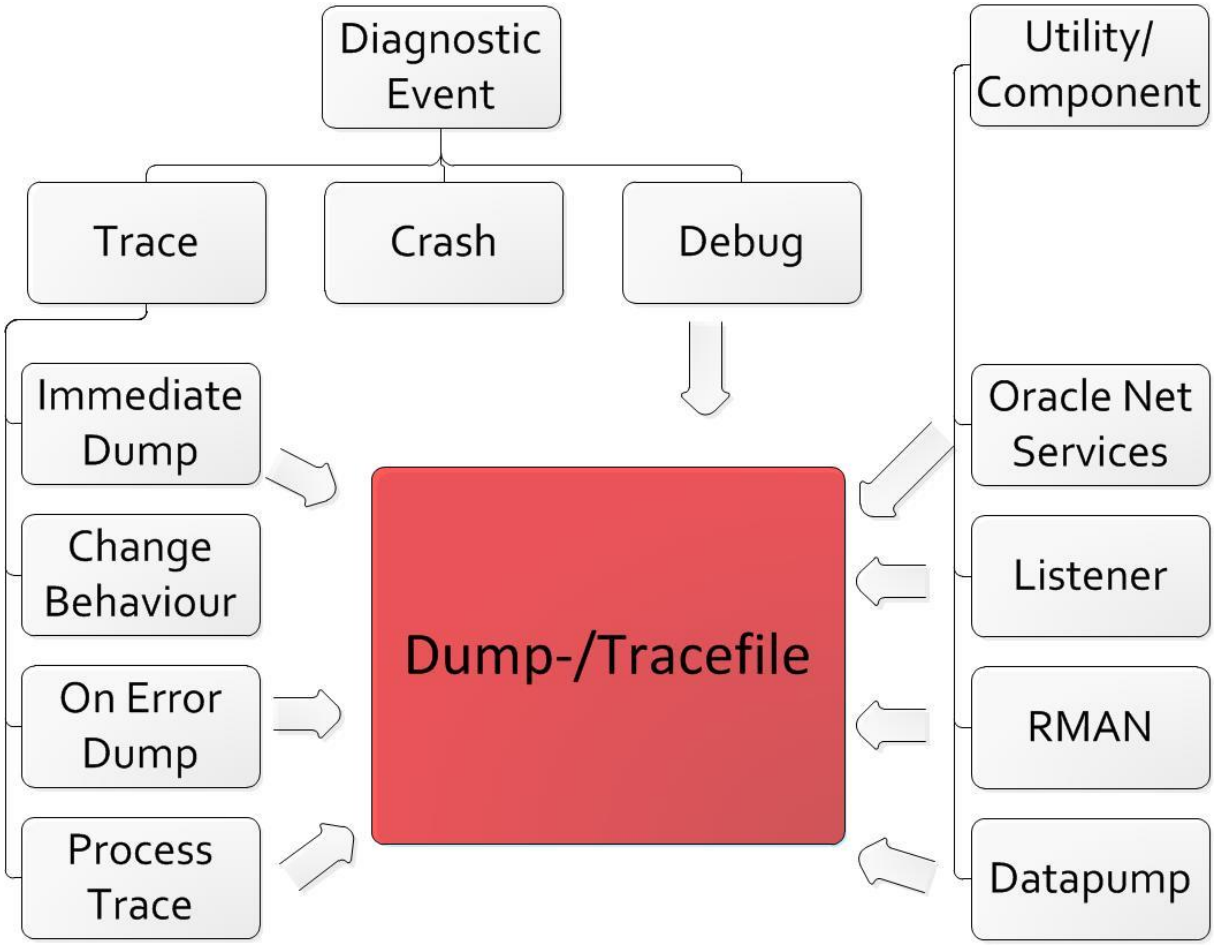

Abb. 1: Quellen von Trace- und Dumpfiles

Unzählige Traces finden sich auch für die Clusterware oder beim Analysieren von Oracle-Datenbankprozessen mit Betriebssystemtools wie pstack oder strace. Auch Entwickler nutzen Traces für das Debugging ihres Codes. Dies sprengt allerdings den Rahmen, so dass an dieser Stelle nicht weiter darauf eingegangen wird.

Events

Traces und Events hängen eng zusammen. Ein grundlegendes Verständnis der Systematik von Events hilft, Tracing passgenau zur Problemstellung einzusetzen. In den offiziellen Dokumentationen von Oracle findet man wenig darüber, was eigentlich genau ein Event ist und welche Events es gibt.

Ein Event ist eine Art „Trigger“ für den Kernel. Bei Auftreten des bezeichneten Events führt er eine bestimmte Aktion aus, die per Default nicht ausgeführt würde. Welche Aktion das sein soll, wird beim Setzen des Events festgelegt. In Abhängigkeit vom Event können verschiedene Aktionen möglich sein:

Aktion nach Eventeintritt	Beispiel
System- oder Prozesstatus zum aktuellen Zeitpunkt in Dumpfile schreiben	Event systemstate
Änderung des Verhaltens der Datenbank	Event 10512: Smon prüft nicht mehr, ob Rollbacksegmente geshrinkt werden können
Errordump bei Auftreten eines Fehlers schreiben	Event 04031: Dumpfile nach Auftreten eines ORA-04031 Fehlers
Fortlaufendes Tracefile schreiben	Event 10046 - SQL_Trace, Tracefile über die Ausführung eines/mehrerer SQL Statements

Events dürfen nie leichtfertig gesetzt werden und sollten mit dem Oracle Support abgestimmt bzw. von ihm angefordert sein.

Jedes Oracle Event hat eine, in manchen Fällen auch mehrere, Nummern. Diese Nummern reihen sich in den Pool der „normalen“ ORA-Errormessages ein und können – zumindest bis einschließlich Oracle Version 12.1.0.2 und auf Nicht-Windows-Systemen – unter $ORACLE_HOME/rdbms/mesg nachgesehen werden.

Das am häufigsten genutzte Event trägt die Nummer 10046. Es dient dem Mitprotokollieren von Laufzeiten, Oracle Wait Events und Execution Plans von SQL Statements. Ebenfalls hin und wieder gesetzt wird Event 10053. Damit wird das Optimizerverhalten getract. Hier kann man wunderbar nachvollziehen, auf welche Weise der Optimizer einen Execution Plan ermittelt, welche Statistiken er heranzieht und welche Joinvarianten getestet und verworfen werden. Weitere Events sind die, die bei ORA-xxxxx-Fehlern die Protokollierung in ein Dumpfile veranlassen. Dies ist sehr hilfreich bei der Analyse der Fehlerursache. Solch ein Event kann für jeden ORA-Fehler gesetzt werden. Die Nummer des Events ist gleich der Fehlernummer – im Beispiel in der obigen Tabelle ist es die 4031.

Aktivierung von Tracing

Eine Möglichkeit, Trace- oder Dumpfiles für die Analyse von Problemen zu gewinnen, ist das Setzen von Events auf der Datenbank selbst. Eine andere Möglichkeit ist das Mitgeben eines TRACE- oder DEBUG-Parameter auf Kommandozeile oder in einer Konfigurationsdatei. Diese Variante ist meist bei kernelferneren Komponenten oder Utilities (zum Beispiel RMAN, Listener oder SQL Net) gegeben.

Manchmal können auch mehrere Möglichkeiten genutzt werden – wie zum Beispiel beim Datapump, dem man sowohl einen TRACE-Parameter beim Aufruf mitgeben kann als auch das Event 39089 in der Datenbank setzen kann, um ein Trace zu erzeugen.

Ein weiteres Beispiel für verschiedene Möglichkeiten, Tracing zu aktivieren, ist das SQL Trace. Hier kann das Event 10046 gesetzt oder – und das ist die aktuell empfohlene Methode – das DBMS_MONITOR Package ausgeführt werden. Zudem gibt es den Parameter set sql_trace=true sowie die Packages DBMS_SYSTEM und DBMS_SUPPORT. Obwohl der Einsatz dieser Parameter/Packages nicht mehr von Oracle empfohlen ist, werden sie doch immer noch vielfältig und problemlos verwendet.

Wider die Datenflut – den Umfang von Tracefiles reduzieren

Tracefiles können aus Tausenden, ja Millionen von Zeilen bestehen. Eine Möglichkeit sich durchzuwuseln ist, von vornherein genau zu überlegen, was die Fragestellung ist. So kann gezielter getract werden.

Tracing kann auf verschiedenen Leveln erfolgen. Es empfiehl sich, mit einem niedrigen Level zu beginnen. Oft ist dies bereits für die Problemanalyse ausreichend. Zudem erleichtert die etwas geringere Datenmenge das Verständnis für den Aufbau des Trace- oder Dumpfiles.

Für die Events 10046 (SQL Trace) und 10053 (Optimizer) ist bereits während des Tracens die Einschränkung auf bestimmte SQL Statements möglich.

Analyse von Tracefiles

Es gibt leider kaum Tools für das Auswerten von Tracefiles. Ausnahme davon sind allein SQL Tracefiles (Event 10046), für die die Tools TKPROF, TRCA und unter bestimmten Gegebenheiten das Tool trcsess die Daten aufbereiten und lesbarer machen.

TKPROF bietet verschiedene Optionen. So können zum Beispiel SYS-Statements rausgefiltert werden. Weiterhin können die im Tracefile enthaltenen Statements nach Laufzeiten, CPU-Verbrauch, physikalischen Plattenzugriffen und ähnlichem sortiert werden. Der Tracefileanalyzer TRCA ist etwas komfortabler – so wird zum Beispiel auch HTML-Output generiert – hierfür müssen allerdings einige Objekte in der Datenbank installiert werden. Trcsess ist ein Tool, das allein zur Konsolidierung SQL Traces aus Shared Server Umgebungen vorgesehen ist. Lt. MOS-Note 280543.1 wird trcsess nicht für den Einsatz von Konsolodierung von Tracesfiles aus parallelen Sessions empfohlen.

Was tun mit all den anderen Tracefiles? Zuallererst: den Mut nicht verlieren! Zum Zweiten: sich Zeit für die Analyse nehmen, das Tracefile erstmal in Ruhe ansehen und `vi` und `grep` einsetzen, um einen Überblick über den Aufbau zu erhalten! Und drittens: Spaß haben beim Entdecken und Erfolg bei der Fehleranalyse!

Quellen:

The full power of Oracle's diagnostic events, part 1 (Tanel Poder)
The full power of Oracle's diagnostic events, part 2 (Tanel Poder)
Diagnostic Events (René. Nyffenegger)
Crash and Debugger Events (Miladin Modrakovic)
MOS-Note 376442.1 - How To Collect 10046 Trace (SQL_TRACE) Diagnostics for Performance Issue

Kontaktadresse:

Katja Werner
Muniqsoft GmbH
Witneystraße, 1
D-82008 Unterhaching

Telefon: +49 (0) 89-6228 6789 0
Fax: +49 (0) 89-6228 6789 50
E-Mail katja.werner@muniqsoft.de
Internet: www.muniqsoft.de