



JDBC Connections & v\$session

Nachbar, Dirk . Senior Consultant . 23. Mai 2008

Gibt es in Unternehmen mehrere J2EE-Anwendungen, die via JDBC-Verbindungen auf eine Oracle Datenbank zugreifen, stellt sich für die IT-Abteilung folgende Frage: Wie stelle ich fest, welche Session zu welcher J2EE-Anwendung gehört? Verwendet man den Oracle Application Server 10.1.3.x, ist es durchaus möglich, diese Informationen sichtbar zu machen.

1. Der Normalfall

In der Regel greifen mehrere J2EE-Anwendungen auf ein Oracle Schema innerhalb der Datenbank zu – meist von einem Oracle Application Server (Oracle AS) aus.

Betrachtet man bei diesem „Normalfall“ einmal den v\$session View, ist man in keiner Weise schlauer.

```
SQL> select username, program, terminal, machine from v$session where username = 'SCOTT';
```

USERNAME	PROGRAM	TERMINAL	MACHINE
SCOTT			pttwinoas
SCOTT			pttwinoas
SCOTT			pttwinoas
SCOTT			pttwinoas
SCOTT			pttwinoas
SCOTT			pttwinoas

6 rows selected.

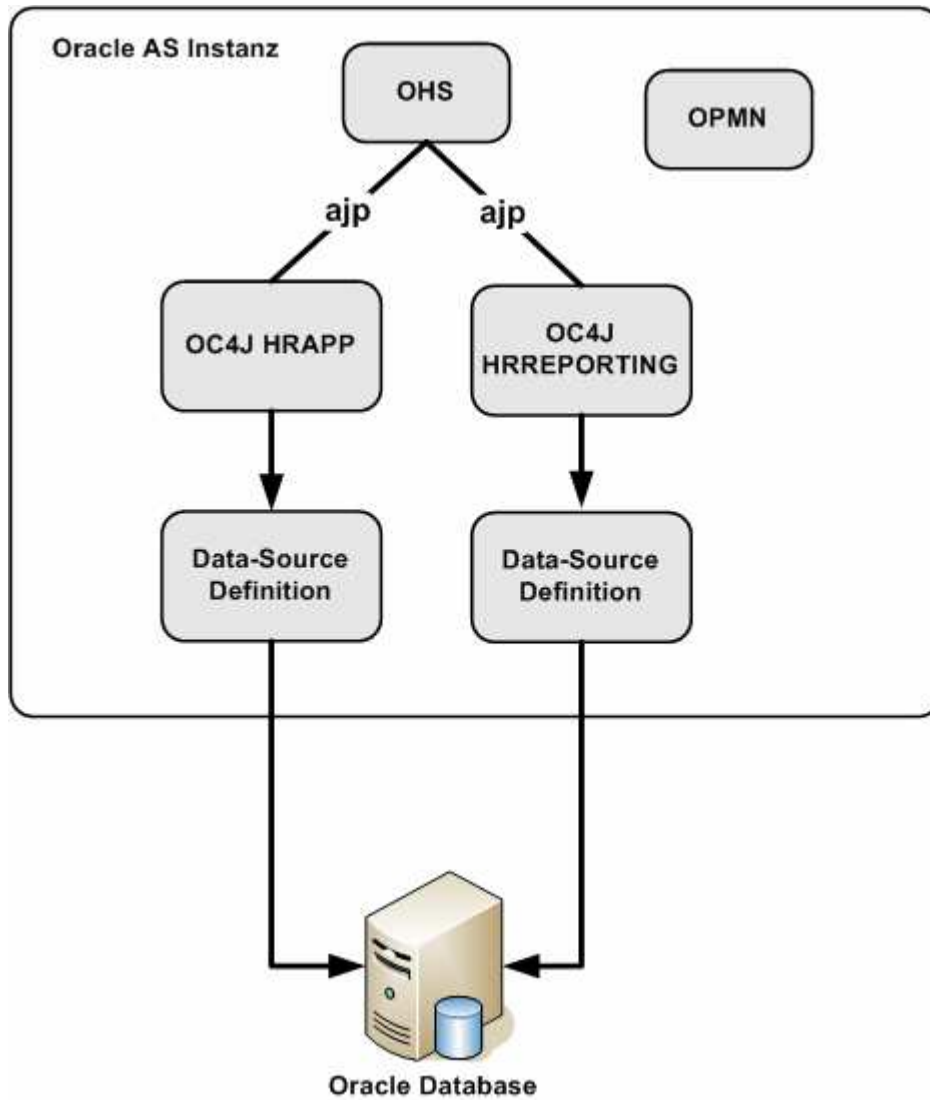
Man sieht lediglich einige Sessions – in diesem Beispiel sechs Sessions des Benutzers SCOTT – und kann den Server identifizieren, auf dem sich der Oracle AS (Machine = pttwinoas) befindet. Man erfährt jedoch nicht, welche Session zu welcher Applikation gehört.

2. Der DBA-freundliche Fall

Verwendet man den Oracle AS 10.1.3.x, braucht man sich keine Gedanken mehr zu machen, wie man die vorhandenen Sessions den jeweiligen J2EE-Anwendungen zuordnet.

Gehen wir von einem Oracle AS aus, auf dem zwei Human Ressource J2EE-Anwendungen mit Zielschema in der Oracle Datenbank laufen, sowie von einem Benutzer mit dem Namen SCOTT.

Im Überblick sieht dieses Szenario wie folgt aus: Die so genannte Oracle AS-Instanz besteht aus einen Oracle HTTP Server (OHS), dem Oracle Process Management & Notification Server (OPMN) sowie einem oder mehreren Oracle Container for J2EE (OC4J) Instanzen – beispielsweise OC4J_HRAPP und OC4J_HRREPORTING. Innerhalb jeder dieser OC4J-Instanzen wird eine Data-Source-Definition angelegt, mit der die Verbindung zur Zieldatenbank definiert wird.



Um die Darstellung des v\$session Views für den DBA freundlicher zu gestalten, muss die Data-Source entsprechend definiert werden.

Jede OC4J-Instanz hat ihre eigene Konfigurationsdatei zur Definition der Data-Source – die so genannte data-source.xml. Diese Konfigurationsdatei findet man unter `$ORACLE_HOME/j2ee/<OC4J_Instancename>/config`.

Im Normalfall sieht die data-source.xml wie folgt aus:



```
<?xml version = '1.0' encoding = 'UTF-8'?>
<data-sources xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation=
    "http://xmlns.oracle.com/oracleas/schema/data-sources-10_1.xsd"
  schema-major-version="10" schema-minor-version="1">

<managed-data-source connection-pool-name="conn_pool_hr_app"
  jndi-name="jdbc/OracleDS" name="OracleDS"/>

<connection-pool name="conn_pool_hr_app"
  initial-limit="3" min-connections="3">
  <connection-factory
    factory-class="oracle.jdbc.pool.OracleDriver"
    user="scott"
    password="tiger"
    url="jdbc:oracle:thin:@//localhost:1521/PROD.ttc.trivadis.com"/>
</connection-pool>
</data-sources>
```

Unter dem Tag <managed-data-source ..> muss man den Data-Source-Namen und die JNDI-Location definieren. Es muss ebenfalls vorgegeben werden, welcher vordefinierte Connection-Pool verwendet werden soll. Unter dem Tag <connection-pool ..> muss die eigentliche Verbindung zur Datenbank definiert werden – beispielsweise wie viele initiale Verbindungen zur Datenbank geöffnet und mit welchem Benutzernamen man verbunden werden soll. Definiert werden muss ebenfalls die eigentliche Datenbankverbindung mit dem Servername, dem Listenerport und dem Servicenamen der Datenbank.

Als DBA kann man nun anhand der v\$session herausfinden, welche Session zu welcher J2EE-Applikation gehört. Hierfür muss man lediglich bei der Definition des Tag <connection-pool ..> wie folgt vorgehen:

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<data-sources xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation=
    "http://xmlns.oracle.com/oracleas/schema/data-sources-10_1.xsd"
  schema-major-version="10" schema-minor-version="1">

<managed-data-source connection-pool-name="conn_pool_hr_app"
  jndi-name="jdbc/OracleDS" name="OracleDS"/>

<connection-pool name="conn_pool_hr_app"
  initial-limit="3" min-connections="3">
  <connection-factory
    factory-class="oracle.jdbc.driver.OracleDriver"
    user="scott"
    password="tiger"
    url="jdbc:oracle:thin:@//localhost:1521/PROD.ttc.trivadis.com">
    <property name="v$session.program" value="conn_pool_hr_app"/>
  </connection-factory>
</connection-pool>
</data-sources>
```

Zuerst muss die factory-class von oracle.jdbc.**pool**.OracleDriver auf oracle.jdbc.**driver**.OracleDriver geändert werden. Der Trailing-Slash am Ende des Tag



<connection-factory muss entfernt werden, und zwar von „url=“jdbc:....PROD.ttc.trivadis.com“/> zu „url=“jdbc:....PROD.ttc.trivadis.com“>. Danach müssen zwei weitere Tags eingefügt werden, einmal <property name..> und abschliessend </connection-factory>.

Über die Value des property name=“v\$session.program“ lässt sich steuern, was künftig in der v\$session unter der Spalte „program“ bei Verbindungen, die über diesen Connection Pool laufen, erscheinen soll.

Jetzt muss man nur noch für die zweite OC4J-Instanz die entsprechende Konfigurationsdatei data-source.xml wie folgt abändern.

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<data-sources xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation=
    "http://xmlns.oracle.com/oracleas/schema/data-sources-10_1.xsd"
  schema-major-version="10" schema-minor-version="1">

<managed-data-source connection-pool-name="conn_pool_hr_reporting"
  jndi-name="jdbc/OracleDS2" name="OracleDS2"/>

<connection-pool name="conn_pool_hr_reporting"
  initial-limit="3" min-connections="3">
  <connection-factory
    factory-class="oracle.jdbc.driver.OracleDriver"
    user="scott"
    password="tiger"
    url="jdbc:oracle:thin:@//localhost:1521/PROD.ttc.trivadis.com">
    <property name="v$session.program" value="conn_pool_hr_app_reporting"/>
  </connection-factory>
</connection-pool>
</data-sources>
```

Nach den Änderungen müssen die OC4J-Instanzen neugestartet werden, damit die neuen Einträge in der Konfigurationsdatei data-source.xml wirksam wird.

Wenn man jetzt eine Abfrage des View v\$session startet, erkennt man sofort, welche Session zu welcher J2EE-Applikation gehört.

```
SQL> select username, program, machine from v$session where username = 'SCOTT' order by program;
```

USERNAME	PROGRAM	MACHINE
SCOTT	conn_pool_hr_app	pttwinoas
SCOTT	conn_pool_hr_app	pttwinoas
SCOTT	conn_pool_hr_app	pttwinoas
SCOTT	conn_pool_hr_app_reporting	pttwinoas
SCOTT	conn_pool_hr_app_reporting	pttwinoas
SCOTT	conn_pool_hr_app_reporting	pttwinoas

6 rows selected.



3. Fazit

Ein schnelles Identifizieren der Sessions ist also mit einfachen Mitteln möglich. Lange „Rätselaktionen“ des DBAs können damit vermieden werden und es bleibt mehr Zeit für die eigentliche Arbeit.

Wie bereits erwähnt ist dieses Feature aber erst ab Oracle AS 10.1.3.x verfügbar. Dies könnte für den ein oder anderen DBA ein Grund sein, jetzt eine Migration zu planen ☺.

Viel Erfolg beim Einsatz von Trivadis Know-how wünscht Ihnen:

Dirk Nachbar

Trivadis SA

Rue Marterey 5

CH-1005 Lausanne

Internet: www.trivadis.com

Tel: +41-21-321 47 00

Fax: +41-21-321 47 01

Mail: info@trivadis.com