# За чем следить и чем управлять при работе приложений с Oracle

Владимир Пржиялковский
Преподаватель технологий Oracle
*prz@yandex.ru*
***www.open-oracle.ru***

Июнь 2008 г.
Небольшая правка: декабрь 2009 г.
Стилистическая правка: май 2012 г.

*… Паллада Афина,*
*За руку взявши, воскликнула к бурному богу Арею:*
*"… не бросим ли мы и троян и ахеян*
*Спорить одних, да Кронид промыслитель их славу присудит ? ... "*
Гомер, Илиада

*Одним из бедствий времени был застарелый произвол доносчиков и их подстрекателей.*
Светоний, Жизнь Двенадцати Цезарей:
Божественный Тит

## Краткое описание

Давнее средство SQL Trace позволяет следить в Oracle за затратами СУБД на обработку запросов SQL серверными процессами, обслуживающими конкретные сеансы связи с СУБД. В версии Oracle 10 появился системный пакет DBMS_MONITOR, разрешающий отслеживать затраты на обработку более разнообразно: к примеру, запросов, принадлежащих определенной службе БД, приложению или его фрагменту, или же запросов, обрабатываемых определенным узлом кластера. В статье показано, как пользоваться этим инструментом наблюдения за обработкой запросов.

## Избирательное слежение за выполнением запросов SQL и загрузкой СУБД средствами пакета DBMS_MONITOR

Давнее средство Oracle SQL Trace позволяет фиксировать «профиль запросов SQL», выдаваемых серверными процессами, обслуживающими сеансы связи с СУБД, и представляет из себя полезный инструмент для выявления проблемных запросов и анализа загрузки СУБД приложениями. До версии 10 оно могло включаться и выключаться только для *конкретных сеансов* связи с СУБД. Это не всегда удобно, поскольку в жизни более насущны профилирование и отладка *приложения*, или даже его фрагментов, а между приложением и сеансом чаще всего нет взаимнооднозначной связи.

Новый для версии 10 пакет DBMS_MONITOR расширил имевшуюся ранее в руках разработчика возможность профилирования SQL-действий сеанса (своего, либо чужого) слежением за действиями отдельных приложений, их групп или частей. Для этого используется модель «служба БД» — «модуль» — «действие». Между этими понятиями нет формальной зависимости, но в жизни предлагается сопоставлять «службе БД» группу логически связанных сеансов приложений, «модулю» — приложение, работающее с данными, а «действию» — эпизод работы приложения. Точнее, средствами пакета можно следить за обработкой запросов SQL с точностью до следующих единиц:

«служб», отождествляемых значением SERVICE_NAME,
«модулей», отождествляемых значением MODULE,
«действий», отождествляемых значением ACTION,
сеансов, отождествляемых значением CLIENT_IDENTIFIER,

сеансов, отождествляемых значениями <*sid*, *serial#*>,
экземпляров СУБД, отождествляемых значением INSTANCE_NAME.

Просмотр и установка единиц слежения рассматриваются ниже.

Сверх того, пакет DBMS_MONITOR позволяет собирать обобщенную статистику выполняемых сеансами или приложениями запросов.

## Единицы слежения

В то время как слежение за действиями *сеанса* и всех *сеансов экземпляра* СУБД не вызывает вопросов, остальные единицы слежения могут требовать пояснения.

Понятие *служба БД* (SERVICE) формировалось в Oracle начиная с версии 8, и к настоящему времени используется для логического группирования сеансов с целью как раз-таки иметь обобщенную единицу слежения и управления при использовании одной БД разными приложениями. Например, сеансы одной и той же службы БД могут присутствовать одновременно на разных узлах кластера в конфигурации RAC, а в рамках одной и той же СУБД могут одновременно исполняться сеансы разных служб. Службу предлагается связывать с набором приложений, объединенных общими свойствами, пороговыми характеристиками или правилами потребления ресурсов СУБД.

Понятие *модуль* (MODULE) используется для обозначения периода работы конкретного приложения в течение сеанса, а понятие *действия* (ACTION) — конкретного фрагмента приложения, то есть эпизода выдачи приложением определенной последовательности команд SQL.

Назначение *отождествителя клиента* (CLIENT IDENTIFIER) — метить сеансы, например установленные по схеме использования совместных серверных процессов (shared server), или же сеансы конечных пользователей при работе через сервер приложений (например, приложений для web).

## Установка единиц слежения и просмотр просмотр выбранных установок

Упомянутые значения единиц слежения за работой приложения наблюдаемы в программе либо из полей таблиц словаря-справочника, либо из встроенного контекста сеанса с именем USERENV:

| ***Значение*** | ***Столбец в таблице*** | ***Параметр контекста сеанса USERENV*** |
|---|---|---|
| SERVICE_NAME | V$SESSION, V$SERVICES | SERVICE_NAME |
| MODULE | V$SESSION | MODULE |
| ACTION | V$SESSION | ACTION |
| CLIENT_IDENTIFIER | V$SESSION | CLIENT_IDENTIFIER |
| SID | V$SESSION | SID |
| SERIAL# | V$SESSION | |
| INSTANCE_NAME | V$INSTANCE | INSTANCE_NAME |

Например, все значения, кроме SERIAL#, наблюдаемы в соответствующих параметрах контекста USERENV сеанса связи клиентской программы с СУБД.

Для дальнейшего просмотра этих параметров из SQL*Plus удобно подготовить файл с параметризованным обращением к контексту USERENV посредством системной функции SYS_CONTEXT:

```
SELECT SYS_CONTEXT ( 'userenv', '&1' ) AS &1 FROM dual
.
SAVE userenv REPLACE
SET VERIFY OFF
```

## Пример установки и просмотра SERVICE_NAME

Значение SERVICE_NAME выставляется автоматически при установлении соединения, а управление службами происходит посредством пакета DBMS_SERVICE.

Простые примеры создания, включения, подсоединения, отключения и удаления:

```
EXECUTE DBMS_SERVICE.CREATE_SERVICE ( 'secunda', 'secunda.class' )
EXECUTE DBMS_SERVICE.START_SERVICE  ( 'secunda' )
CONNECT /@localhost:1521/secunda.class AS SYSDBA
EXECUTE DBMS_SERVICE.STOP_SERVICE   ( 'secunda' )
EXECUTE DBMS_SERVICE.DELETE_SERVICE ( 'secunda' )
```

Все операции, кроме подсоединения к СУБД, выполняет администратор разовым порядком. Он же может использовать еще один способ управления службами — через программу *srvctl*.

Значения SERVICE_NAME указываются в установках Oracle Net, регистрируются процессом *listener* и наблюдаются через программу *lsnrctl*. Значения зарегистрированных в рамках БД служб наблюдаются к тому же в таблице V$SERVICES в несколько расширенном составе. Так, помимо сконфигурированных администратором, всегда имеются две внутренние, несетевые, службы: SYS$BACKGROUND используется для автоматической приписки фоновых процессов СУБД, а SYS$USERS — серверных процессов местных (локальных) подключений. Служба primaXPT, присутствовавшая в текстах примеров выше, обязана наличию в использовавшейся БД инфраструктуры XML DB.

Пример просмотра:

```
SQL> CONNECT scott/tiger@prima.class
Connected.
SQL> @userenv SERVICE_NAME

SERVICE_NAME
----------------------------------------------------------------
prima.class
```

*Упражнение*. Просмотреть от имени администратора список служб, зарегистрированных для рабочей БД. Просмотреть список сеансов через таблицу V$SESSION с выдачей принадлежности процессов службам БД. Объяснить наличие процессов в рамках служб SYS$USERS и SYS$BACKGROUND.

## Пример установки и просмотра MODULE и ACTION

Значения MODULE и ACTION устанавливаются в программе с помощью пакета DBMS_APPLICATION_INFO. Пример:

```
SQL> EXECUTE DBMS_APPLICATION_INFO.SET_MODULE -
  2           ( 'some module', 'some action' )

PL/SQL procedure successfully completed.

SQL> @userenv MODULE

MODULE
----------------------------------------------------------------
some module

SQL> @userenv ACTION

ACTION
----------------------------------------------------------------
some action

SQL> EXECUTE DBMS_APPLICATION_INFO.SET_ACTION ( '' )

PL/SQL procedure successfully completed.
```

```
SCOTT> /

ACTION
------------------------------------------------------------------------


SQL>
```

### Пример установки и просмотра CLIENT_IDENTIFIER

Значение CLIENT_IDENTIFIER выставляется программно с помощью пакета DBMS_SESSION. Пример:

```
SQL> @userenv CLIENT_IDENTIFIER

CLIENT_IDENTIFIER
------------------------------------------------------------------------


SQL> EXECUTE DBMS_SESSION.SET_IDENTIFIER ( 'Web client' )

PL/SQL procedure successfully completed.

/

CLIENT_IDENTIFIER
------------------------------------------------------------------------
Web client
```

### Примеры отслеживания запросов SQL со стороны приложения и его фрагментов

Процедура SERV_MOD_ACT_TRACE_ENABLE из пакета DBMS_MONITOR позволяет следить за выдачей запросов SQL отдельными «приложениями», «модулями» в составе приложений и «действиями» в рамках модулей. Пример:

```
DBMS_MONITOR.SERV_MOD_ACT_TRACE_ENABLE (
  service_name  => 'prima.class'
, module_name   => 'SQL*Plus'
, action_name   => DBMS_MONITOR.ALL_ACTIONS
, waits         => TRUE
, binds         => TRUE
, instance_name => NULL
);
/* ... Включили наблюдение.
      Приложения работают, СУБД снимает показания работы ...
      Завершаем наблюдение:
*/
DBMS_MONITOR.SERV_MOD_ACT_TRACE_DISABLE (
  service_name  => 'prima.class'
, module_name   => 'SQL*Plus'
, action_name   => DBMS_MONITOR.ALL_ACTIONS
, instance_name => NULL
);
```

Для параметра MODULE_NAME по аналогии можно указать ALL_MODULES.

Таким же путем включается отслеживание запросов, выполняемых текущим или чужими сеансами (SESSION_TRACE_ENABLE/DISABLE), сеансами, помеченными заданым значением CLIENT_IDENTIFIER (CLIENT_ID_TRACE_ENABLE/DISABLE), а также всеми сеансами подряд или же порождаемыми только определенным экземпляром СУБД, например, в конфигурации RAC (DATABASE_TRACE_ENABLE/DISABLE).

Пример начала и прекращения отслеживания выполнения запросов сеансами, помеченными

CLIENT_IDENTIFIER = 'Web client':

```
DBMS_MONITOR.CLIENT_ID_TRACE_ENABLE ( client_id => 'Web client' )
/*  ... Включили наблюдение.
    Приложения работают, СУБД снимает показания работы ...
    Завершаем наблюдение:
*/
DBMS_MONITOR.CLIENT_ID_TRACE_DISABLE ( client_id => 'Web client' )
```

Просмотр профиля выдачи запросов SQL выполняется с помощью программ *tkprof* и *trcsess*. Последняя программа стала поставляться в ПО версии Oracle 10, и как раз-таки и выбирает из трассировочных файлов разных процессов данные, относящиеся к указанным единицам слежения. Вот пример, как может выглядеть совместное использование этих программ:

```
>trcsess output=out.txt module=SQL*Plus c:\oracle\admin\prima\udump\*
>tkprof out.txt final.txt
```

Еще пример сведения воедино накопленных данных:

```
>trcsess output=out.txt clientid="Web client" c:\oracle\admin\prima\udump\*
```

*Упражнение*. Включить отслеживание всех запросов SQL, поступающих от сеансов SQL*Plus, и наблюдать накопление профиля выдачи запросов. Выключить отслеживание.

## Пример сбора статистики о запросах SQL в приложении

Для сеансов, помеченых CLIENT_IDENTIFIER, и для приложений, помеченых комбинациями SERVICE_NAME — MODULE — ACTION, пакет DBMS_MONITOR позволяет не только собирать профиль выдачи запросов SQL, но и статистику затрат СУБД на обработку запросов.

Пример со сбором статистики о работе определенных клиентов. Выдадим:

```
CONNECT / as sysdba
EXECUTE DBMS_MONITOR.CLIENT_ID_STAT_ENABLE ( 'Web client' )
HOST sqlplus scott/tiger
EXECUTE DBMS_SESSION.SET_IDENTIFIER ( 'Web client' )
SELECT COUNT ( * ) FROM dual;
EXIT
```

Здесь с помощью команды HOST программы SQL*Plus был осуществлен короткий запуск сеанса, помеченного значением CLIENT_IDENTIFIER = 'Web client'.

Продолжим в изначальном сеансе от имени SYS:

```
SYS> SELECT aggregation_type, primary_id
  2> FROM dba_enabled_aggregations;

AGGREGATION_TYPE      PRIMARY_ID
--------------------- ----------------------------------------------
CLIENT_ID             Web client

SYS> COLUMN client_identifier FORMAT A20
SYS> COLUMN stat_name         FORMAT A35
SYS> COLUMN value             FORMAT 99999999
SYS> SELECT client_identifier, stat_name, value
  2> FROM v$client_stats
  3> ;

CLIENT_IDENTIFIER    STAT_NAME                                VALUE
-------------------- ---------------------------------- ---------
Web client           user calls                               4
Web client           DB time                               3807
Web client           DB CPU                                3807
```

```
Web client           parse count (total)                      2
Web client           parse time elapsed                     421
Web client           execute count                            4
Web client           sql execute elapsed time              1940
Web client           opened cursors cumulative                2
...

SYS> EXECUTE DBMS_MONITOR.CLIENT_ID_STAT_DISABLE ( 'Web client' )

PL/SQL procedure successfully completed.
```

Вот перечень системных таблицы, позволяющих контролировать сбор статистики и наблюдать результаты:

DBA_ENABLED_AGGREGATIONS
V$CLIENT_STATS
V$SERVICE_STATS
V$SERV_MOD_ACT_STATS
V$SERVICEMETRIC
V$SERVICEMETRIC_HISTORY

Последние две дают подробную информацию о расходовании процессорного времени.

Это был пример с использованием процедур **CLIENT_ID_STAT**_ENABLE/DISABLE. Процедуры **SERV_MOD_ACT**_STAT_ENABLE/DISABLE используются аналогично.

*Упражнение*. Включить сбор обобщенной статистики выполнения запросов SQL всех соединений к СУБД по SQL*Plus и наблюдать ее накопление. Отключить сбор статистики.

### Не только слежение, и не только из программы

Из указаных выше единиц слежения наиболее разработанной является служба. Она не только позволяет следить за действиями приложений, но и является единицей управления. Вот пример путей ее употребления в этом качестве на настоящий момент:

- В планировщике заданий со службой можно связать конкретный класс заданий (пакет DBMS_SCHEDULER, процедура CREATE_JOB_CLASS).
- Сеансы, осуществляемые в рамках службы, можно указать в определении группы потребителей ресурсов СУБД при работе с распределителем (диспетчером) ресурсов (пакет DBMS_RESOURCE_MANAGER, процедура SET_CONSUMER_GROUP_MAPPING).
- В системе предупредительной сигнализации именно для конкретной службы можно устанавливать пороговые значения для общего времени (ELAPSED_TIME_PER_CALL) и процессорного времени (CPU_TIME_PER_CALL) обработки запросов SQL, о превышении которых СУБД будет докладывать автоматически (пакет DBMS_SERVER_ALERT, процедура SET_THRESHOLD).
- Процедура DISCONNECT_SESSION из пакета DBMS_SERVICE останавливает все сеансы требуемой службы.

Упомянутые пакеты DBMS_SCHEDULER, DBMS_SERVER_ALERT и DBMS_SERVICE, а также процедура SET_CONSUMER_GROUP_MAPPING старого пакета DBMS_RESOURCE_MANAGER появились в версии 10.

Кроме того, все реализованные программно возможности слежения и управления активно представлены соответствующими web-страницами OEM.