

## Как отслеживать работу приложений и их составных частей в Oracle

(Defining a Cascading List of Values by Steve Muench )

Владимир Пржиляковский, Преподаватель технологий Oracle  
prz@yandex.ru, <http://www.ccas.ru/prz>

Источник: Статья предоставлена автором, январь 2008 г.



*Одним из бедствий времени был застарелый произвол  
доносчиков и их подстрекателей.*

*Светоний “Жизнь Двенадцати Цезарей:  
Божественный Тит”*

*Затем Цезарь оглашает доносы и вызывает свидетелей ...  
Тацит “Анналы”*

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. До версии 10 оно могло включаться и выключаться только для конкретных сеансов связи с СУБД. Это далеко 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 ранее имевшуюся возможность трассировки действий в рамках сеанса (своего или чужого) возможностью отслеживания действий отдельных приложений и их частей. Для последней цели используется модель «служба БД» - «модуль» - «действие». Между этими понятиями нет формальной зависимости, но методически предлагается связывать с «модулем» приложение, работающее с данными, предоставляемыми «службой БД», а с «действием» - работу фрагментов приложения. Точнее, средствами пакета можно отслеживать выдачу запросов SQL со стороны следующих единиц:

- \* «служб», отождествляемых значением SERVICE\_NAME,
- \* «модулей», отождествляемых значением MODULE,
- \* «действий», отождествляемых значением ACTION,
- \* сеансов, отождествляемых значением CLIENT\_IDENTIFIER,
- \* сеансов, отождествляемых значениями ,
- \* экземпляров СУБД, отождествляемых значением

INSTANCE\_NAME.

Просмотр и установка единиц слежения рассматриваются ниже.

Вдобавок пакет DBMS\_MONITOR позволяет собирать обобщенную статистику выполняемых сеансами или приложениями запросов.

Установка единиц слежения для DBMS\_MONITOR и просмотр существующих значений

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

Значение/	Поле таблиц	П а р а м е т р контекста
USERENV сеанса		
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 с СУБД.

Для дальнейшего просмотра этих параметров в SQL\*Plus удобно подготовить файл 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
```

### Установка и пример просмотра SERVICE\_NAME

Значение SERVICE\_NAME выставляется автоматически при установлении соединения или программно с помощью вызова DBMS\_SERVICE.CREATE\_SERVICE. Пример просмотра:

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

```
SQL> @userenv SERVICE_NAME
```

```
SERVICE_NAME
```

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

Возможные значения SERVICE\_NAME указываются в сетевых установках Oracle и должны быть зарегистрированы в качестве службы БД процессом listener. Значения зарегистрированных в рамках БД служб можно наблюдать в таблице V\$SERVICES.

**Упражнение.** Просмотреть список служб, зарегистрированных для рабочей БД. Дать попытку интерпретации служб SYSS\$USERS и SYSS\$BACKGROUND. Проверить предположение.

### Пример установки и просмотра MODULE и ACTION

Значения MODULE и ACTION устанавливаются программно 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 или же вызовом OCI setClientIdentifier (только JDBC в версии 9). Пример:

```
SQL> @userenv CLIENT_IDENTIFIER
```

```
CLIENT_IDENTIFIER
```

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

```
PL/SQL procedure successfully completed.
```

```
/
```

```
CLIENT_IDENTIFIER
```

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

Назначение CLIENT\_IDENTIFIER - «метить» сеансы конечных пользователей, организованные по схеме использования общих серверных процессов (shared server), в частности, через сервер приложений (например, приложений для web).

**Пример отслеживания запросов SQL со стороны приложения и его элементов**

Процедура SERV\_MOD\_ACT\_TRACE\_ENABLE 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
```

## Современные технологии | Как отслеживать работу приложений и их составных частей в Oracle

```
, binds => TRUE
, instance_name => NULL
);
/*
* выжидаем, пока продолжается работа
*/
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 образом включается трассирование запросов, поступающих из текущего или чужого сеансов (SESSION\_TRACE\_ENABLE/DISABLE), сеансов, помеченных CLIENT\_IDENTIFIER (CLIENT\_ID\_TRACE\_ENABLE/DISABLE) и из всех сеансов подряд, или же порождаемых определенным экземпляром СУБД, например из конфигурации RAC (DATABASE\_TRACE\_ENABLE/DISABLE).

Просмотр профиля выдачи запросов SQL выполняется с помощью программ tkpro/ 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
```

**Упражнение.** Включить трассирование всех запросов SQL, поступающих от SQL\*Plus, и наблюдать накопление профиля выдачи запросов. Отключить трассирование.

Пример сбора статистики о запросах SQL в приложении

Для сеансов, помеченных CLIENT\_IDENTIFIER, и для приложений, помеченных 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 cli-
ent')
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.