INSTRUMENTATION!
INSTRUMENTATION!
INSTRUMENTATION!

David Kurtz

UKOUG 2025

/*+Go-Faster*/ Consultancy

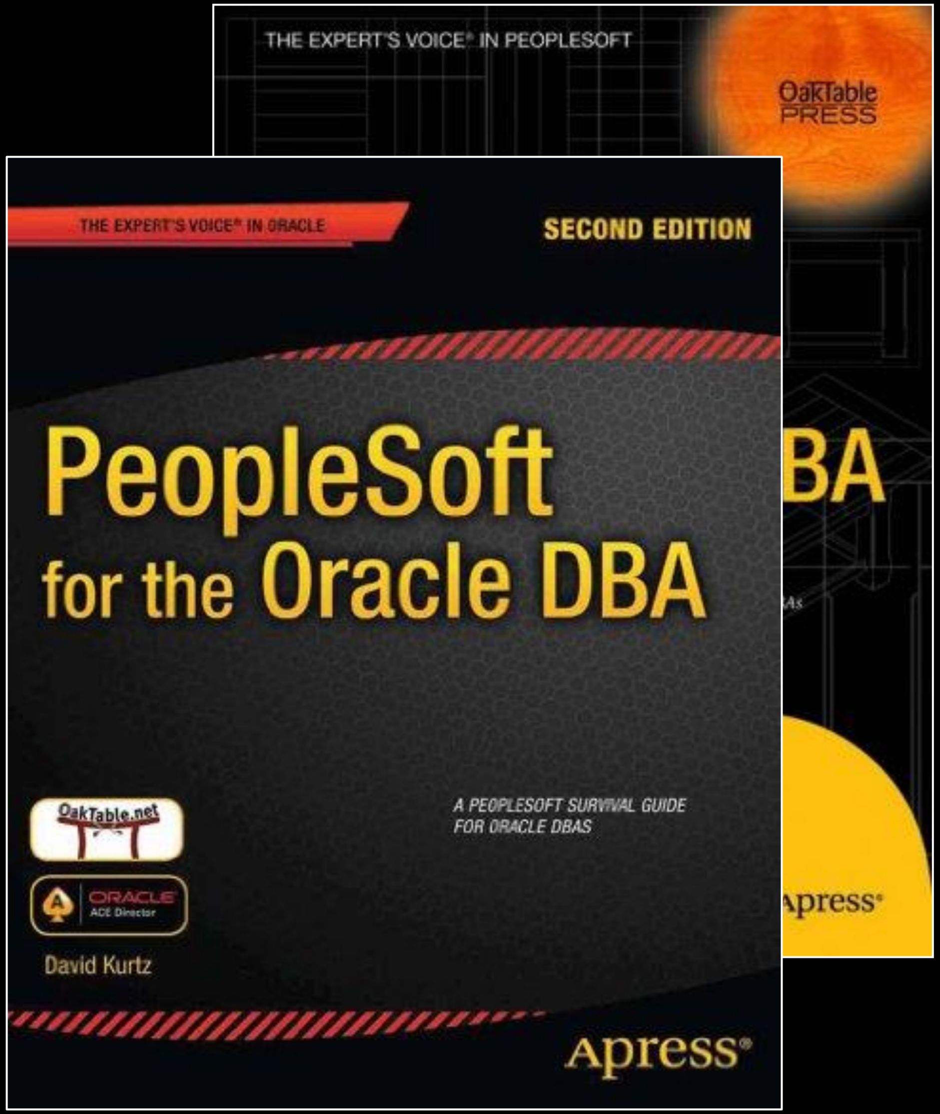
WHO AM I

Go-Faster Consultancy

- Performance tuning
  - Oracle RDBMS
  - PeopleSoft ERP
- www.go-faster.co.uk
- blog.go-faster.co.uk
- blog.psftdba.com

OakTable.net Oak Table

ORACLE ACE Director



INSTRUMENTATION

Implemented by Developers

- **Added to code**
- **Identify the program/procedure/function that is running**

Oracle session tracks

- **Database User, Service, Program, OS User, Machine ID, Client ID**
- **Transaction**
- **Execution Context ID (ECID)**

Additional Instrumentation

- **DBMS_APPLICATION_INFO package**
  - **Module/Action → ASH**
  - **CLIENT_INFO**

Used by Administrators

- **Performance is all about time**
  - **How long did the application spend doing what?**
  - **Find what it spent the most time on and address that first.**
  - **Whatever you are spending time on, stop doing it.**
  - **In the cloud, you mostly pay for time spent on CPU.**
- **Instrumentation tells you where you are in an application**
  - **Now you know where time is spent in the application.**

INSTRUMENTATION –V- LOGGING/TRACE

Instrumentation is held on the session

- Read by database monitoring (Diagnostics pack)
  - AWR (SQL Statistics)
  - ASH
- Gives context to those metrics
- Overhead?
  - Minimal runtime overhead
  - Possibly negative, because you are blind without it

Logging records that something happened

- When it happened
- Some context about it

Session Trace

- SQL Statement with bind Variable
- Execution Plan
- Duration of each execute/fetch
- Detail of each wait by event
- MODULE/ACTION
  - Initial Value / Each new value
- Measurement Intrusion

MODULE & ACTION

Sessions are tracked via SGA memory structure

- exposed as *sys.x$ksuse*

Module: *ksuseapp*

- defaults to program name for foreground processes
- Some background processes area instrumented: KTSJ

Action: *ksuseact* - defaults to null

64-character registers

Exposed via

- v$session

Transferred from v$session

- x$kglcursor_child => v$sql
- x$kglcursor_child_sqlid => v$sqlarea

AWR/ASH

- v$active_session_history
- dba_hist_sqlstat
- dba_hist_Active_Session_history

Also

- SQL Plan Baselines
- SQL Tuning
- Database Replay (FKA RAT)
- SQL Translation Framework

TRACE BY SERVICE/MODULE/ACTION

Enable Session Trace for specific values

- Like Watchpoint in a Debugger
- For specific services, module, action
  - Equality matching only

- dbms_monitor.serv_mod_act_trace_disable()
- SYS.WRI$_TRACING_ENABLED

Database will enable/disable session trace

- as session moves to/from matching values

Module/Action is recorded in trace

Some trace profilers filter report/filter by module

- ORASRP
- Not TKPROF

Trcsess utility

- Merge traces
- filter by module/action

https://blog.yannickjaquier.com/oracle/trace-analyzer-vs-orasrp-vs-oracle-trace-parsing-vs-trivadis-extended-tracefile-analysis.html

TRACE BY SERVICE/MODULE/ACTION

```
ALTER SESSION SET tracefile_identifier = 'MY_TRACE_DEMO';
exec DBMS_MONITOR.SERV_MOD_ACT_TRACE_ENABLE -
(service_name=>'mydb', module_name=>'MY_MODULE', action_name=>'MY_ACTION');


DECLARE
l_dummy  VARCHAR2(10);
BEGIN
  dbms_application_info.set_module(module_name=>'MY_MODULE', action_name=>'MY_ACTION');

  SELECT dummy INTO l_dummy FROM DUAL;

  dbms_application_info.set_module(module_name=>'SQL*Plus', action_name=>'');
END;
/
```

TRACE BY SERVICE/MODULE/ACTION

```
Trace file /u01/app/oracle/diag/rdbms/mydb_8zp_fra/mydb1/trace/mydb1_ora_85058_MY_TRACE_DEMO.trc
Oracle Database 19c EE Extreme Perf Release 19.0.0.0.0 - Production
Version 19.28.0.0.0
…
*** 2025-11-28T14:30:34.286280+00:00 (mydb(4))
*** SESSION ID:(42.53053) 2025-11-28T14:30:34.286306+00:00
*** CLIENT ID:() 2025-11-28T14:30:34.286310+00:00
*** SERVICE NAME:(mydb) 2025-11-28T14:30:34.286313+00:00
*** MODULE NAME:(MY_MODULE) 2025-11-28T14:30:34.286317+00:00
*** ACTION NAME:(MY_ACTION) 2025-11-28T14:30:34.286320+00:00
*** CLIENT DRIVER:(jdbcthin : 21.1.0.0.0) 2025-11-28T14:30:34.286323+00:00
*** CONTAINER ID:(4) 2025-11-28T14:30:34.286326+00:00
=====================
PARSING IN CURSOR #139817447210152 len=22 dep=1 uid=0 oct=3 lid=0 tim=109220247162 hv=2186682389
ad='91e19550' sqlid='47pmdpk15c80p'
SELECT DUMMY FROM DUAL
END OF STMT
PARSE #139817447210152:c=34,e=34,p=0,cr=0,cu=0,mis=0,r=0,dep=1,og=1,plh=272002086,tim=109220247158
EXEC #139817447210152:c=29,e=29,p=0,cr=0,cu=0,mis=0,r=0,dep=1,og=1,plh=272002086,tim=109220247533
FETCH #139817447210152:c=53,e=53,p=0,cr=2,cu=0,mis=0,r=1,dep=1,og=1,plh=272002086,tim=109220247609
STAT #139817447210152 id=1 cnt=1 pid=0 pos=1 obj=143 op='TABLE ACCESS FULL DUAL (cr=2 pr=0 pw=0 str=1
time=49 us cost=2 size=2 card=1)'
CLOSE #139817447210152:c=0,e=0,dep=1,type=3,tim=109220247708
```

RESOURCE MANAGER CONSUMER GROUP MAPPING

Resource Manager (EE only feature)

- Consumer group mappings to different session attributes.
  - [SERVICE,]MODULE[,ACTION]
  - User, Program
- User defined hierarchy of attribute mappings
- Explicitly
  - Scheduler job class

- Attribute value matching
  - =
  - LIKE
  - REGEXP

SAMPLE RESOURCE PLAN FOR PEOPLESOFT

Consumer Group	Management Priority	CPU Guarantee	Limits
SYS_GROUP	1	100%	
PSFT_GROUP	2	79%	PQ limit = CPU_COUNT
PSFT_HIGHPQ_GROUP	2	20%	PQ limit = CPU_COUNT
HIGH_GROUP	2	1%	PQ limit = CPU_COUNT
SUML_GROUP	3	100%	PQ limit = CPU_COUNT
PSFT_BATCH	4	100%	PQ limit = CPU_COUNT
NVISION_GROUP	5	100%	PQ limit = CPU_COUNT
PSQUERY_ONLINE_GROUP	6	90%	20-minute limit
PSQUERY_BATCH_GROUP	6	1%	4-hour limit
NVSRUN_GROUP	6	9%	PQ limit = CPU_COUNT
LOW_GROUP	8	1%	
LOW_LIMITED_GROUP	8	1%	2-hour limit, PQ limit = CPU_COUNT
OTHER_GROUPS	8	1%	PQ limit = CPU_COUNT

DECLARATIVELY MAP INSTRUMENTATION → CONSUMER GROUP

```
DBMS_RESOURCE_MANAGER.set_consumer_group_mapping
(attribute => DBMS_RESOURCE_MANAGER.CLIENT_PROGRAM
, value => 'PSAPPSRV%'
, consumer_group => 'PSFT_GROUP');

DBMS_RESOURCE_MANAGER.set_consumer_group_mapping
(attribute => DBMS_RESOURCE_MANAGER.MODULE_NAME_ACTION
, value => 'QUERY_MANAGER.QUERY_VIEWER'
, consumer_group => 'PSQUERY_ONLINE_GROUP');

DBMS_RESOURCE_MANAGER.set_consumer_group_mapping
(attribute => DBMS_RESOURCE_MANAGER.MODULE_NAME
,value => 'PSAE.PSQUERY.%'
,consumer_group => 'PSQUERY_BATCH_GROUP');

DBMS_RESOURCE_MANAGER.set_consumer_group_mapping
(attribute => DBMS_RESOURCE_MANAGER.MODULE_NAME_ACTION
, value => 'RPTBOOK.PI=%:USGL234X:%'
, consumer_group => 'HIGH_GROUP');
```

PL/SQL INSTRUMENTATION

```
k_module      CONSTANT VARCHAR2(64) := $$PLSQL_UNIT;
…
PROCEDURE my_procedure IS
  l_module VARCHAR2(64);
  l_action VARCHAR2(64);
BEGIN
  dbms_application_info.read_module(module_name=>l_module
                                    ,action_name=>l_action);
  dbms_application_info.set_module(module_name=>k_module
                                  ,action_name=>'MY_PROCEDURE');
…
  dbms_application_info.set_module(module_name=>l_module
                                  ,action_name=>l_action);
EXCEPTION
  WHEN … THEN
    dbms_application_info.set_module(module_name=>l_module
                                    ,action_name=>l_action);
    RAISE / EXIT
…
END my_procedure;
```

In PL/SQL

- MODULE = package name
- ACTION = procedure/function name

DIY INSTRUMENTATION

You may need to be creative!

- PeopleTools: When a process starts (and sets its own status), I have a trigger that sets module and action

```
CREATE OR REPLACE TRIGGER sysadm.psftapi_store_prcsinstance
BEFORE UPDATE OF runstatus ON sysadm.psprcsrqst FOR EACH ROW
WHEN ((new.runstatus IN('3','7','8','9','10') OR old.runstatus IN('7','8')) AND new.prcstype != 'PSJob')
BEGIN
…
  psftapi.set_action(p_prcsinstance=>:new.prcsinstance
   ,p_runstatus=>:new.runstatus, p_prcsname=>:new.prcsname);
…
EXCEPTION WHEN OTHERS THEN NULL; --do not crash the scheduler
END;
/
```

DIY INSTRUMENTATION
FINE-GRAINED AUDIT & HANDLER

FGA Policy on query run control table

- table passes parameters into process
- 0 days purge policy

```
BEGIN
  DBMS_FGA.ADD_POLICY(
   object_schema   => 'SYSADM',
   object_name     => 'PS_NVS_REPORT',
   policy_name     => 'PS_NVS_REPORT_SEL',
   handler_module  => 'GFC_FGA_NVISION_HANDLER',
   enable          =>  TRUE,
   statement_types=> 'SELECT',
   audit_trail     => DBMS_FGA.DB + DBMS_FGA.EXTENDED
  );
END;
/
```

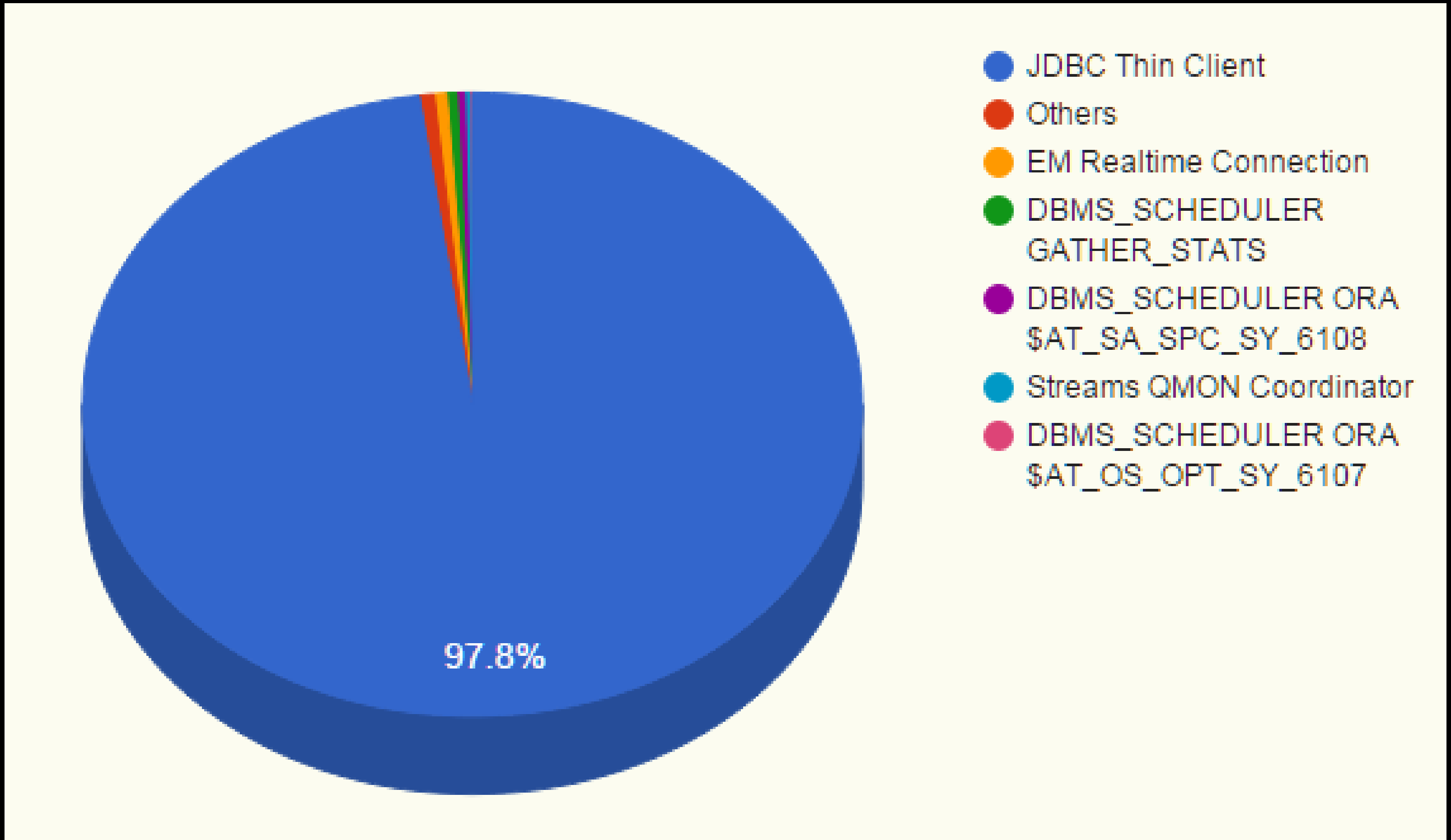
FGA Audit Handler

- PL/SQL to update ACTION

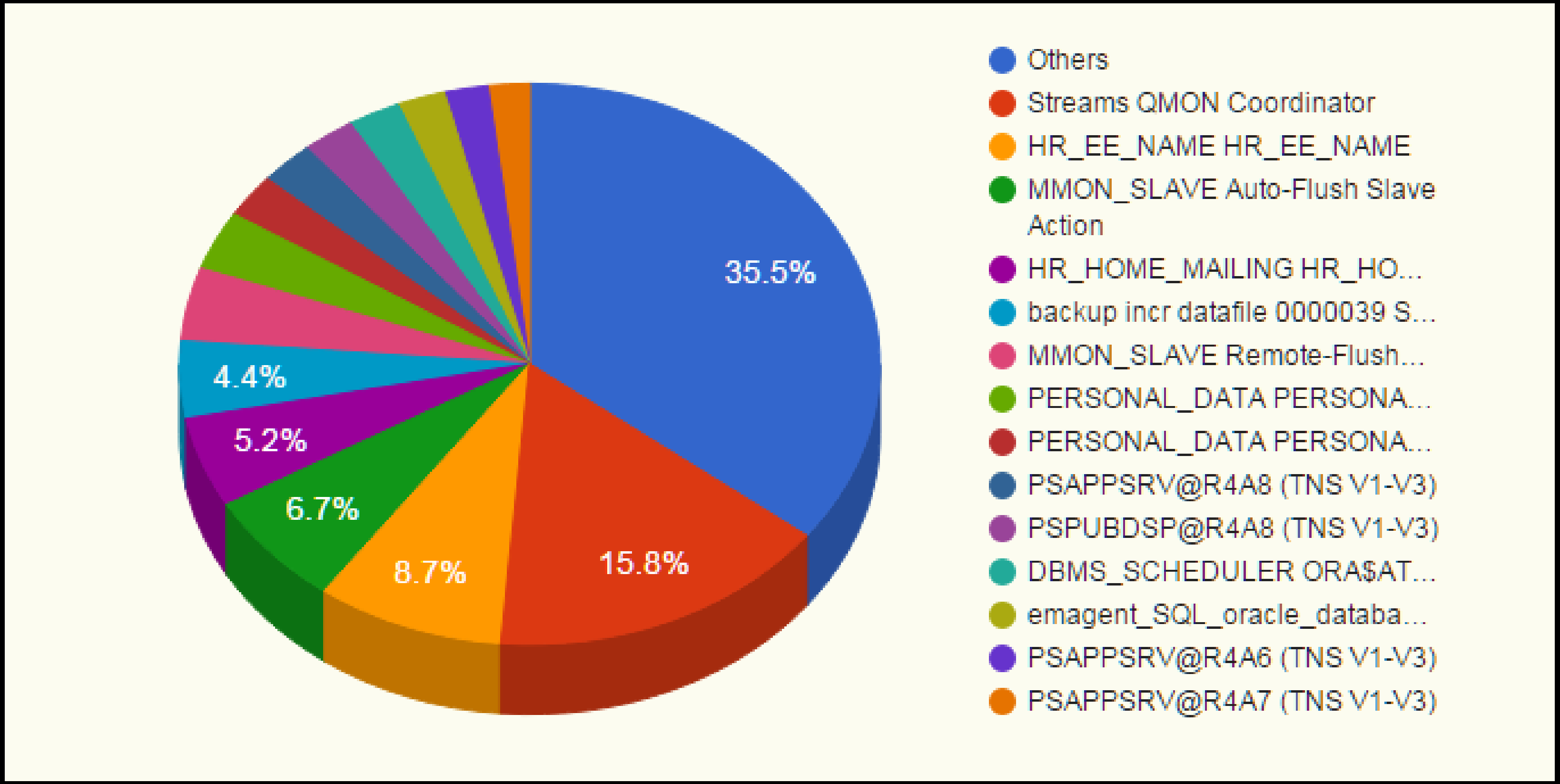
```
CREATE PROCEDURE sysadm.gfc_fga_nvision_handler
(object_schema VARCHAR2
,object_name   VARCHAR2
,policy_name   VARCHAR2) AS …
BEGIN
…
  dbms_application_info.set_action(…);
END;
/
```

https://blog.psftdba.com/2017/11/nvision-performance-tuning-5-additional.html

NO INSTRUMENTATION!



WITH INSTRUMENTATION!



GUIDANCE FOR INSTRUMENTATION

MODULE/ACTION are just character strings

- VARCHAR2(64)

Recommendation:

- Think of MODULE as the parent of ACTION
  - Eg. package and procedure
- Be consistent across your application

Variable values in MODULE/ACTION

- Can improve understanding of instrumentation.
- Can identify specific instance of a process
- May need to replace them in scripts to profile
- Consumer Group mappings can pattern match
- SQL Trace
  - By MODULE [and ACTION]
    - Like watchpoint in debugger
  - Exact match only
    - https://blog.psftdba.com/2012/09/oracle-sql-tracing-by-module-action.html

SQL TO SIMPLIFY INSTRUMENTATION PROFILING WITH VARIABLES

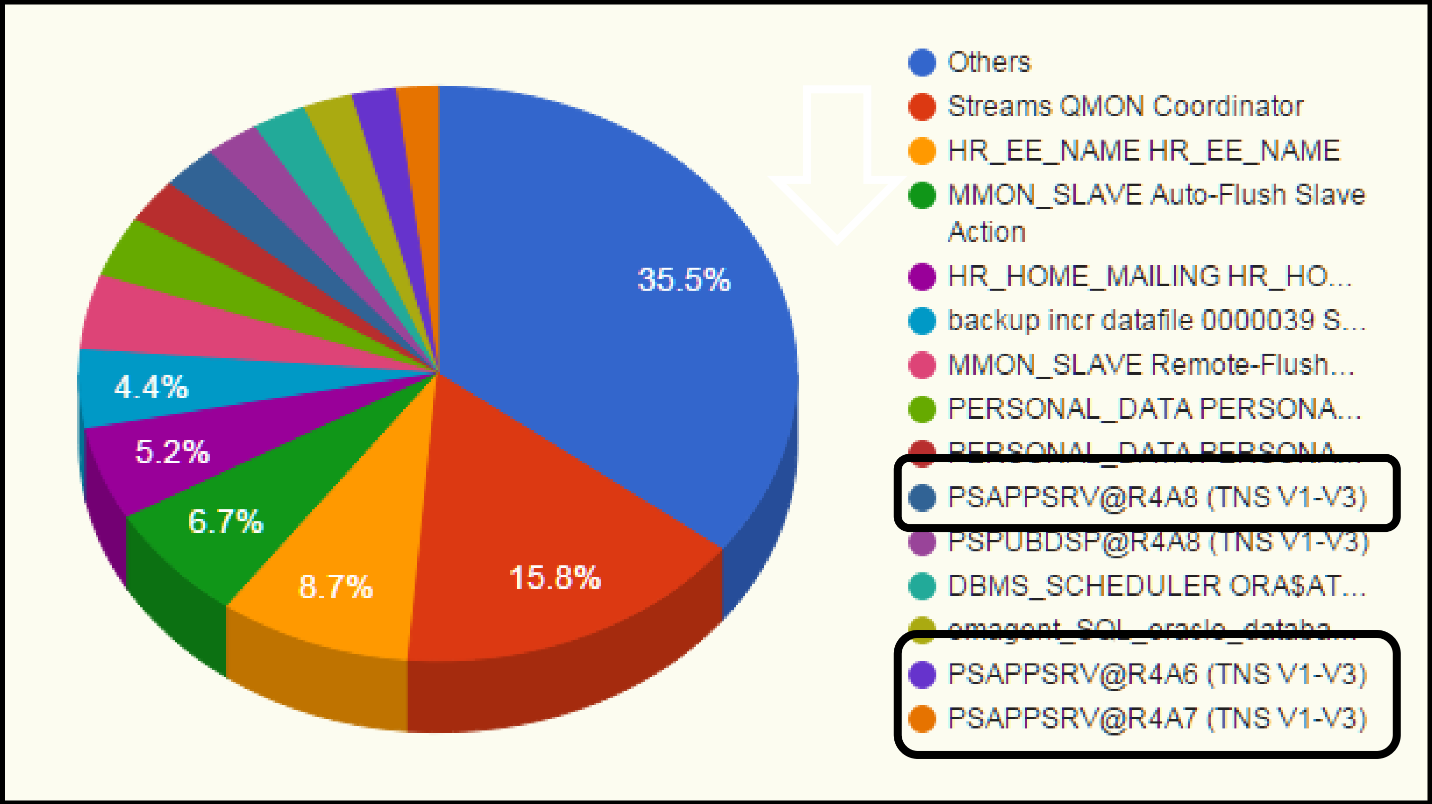
It depends on how the application is instrumented.

replace() **or** ***regexp_replace()***

Module defaults to program

- **Strip after @ or after .**
- ***REGEXP_SUBSTR(h.module, '[^.@]+',1,1)***

- ***PSAPPSRV@RA46 {TNS V1-V3)***
- ***PSAPPSRV@RA47 {TNS V1-V3)***
- ***PSAPPSRV@RA48 {TNS V1-V3)***
  - ***→ PSAPPSRV***



SQL TO SIMPLIFY INSTRUMENTATION PROFILING WITH VARIABLES

It depends on how the application is instrumented.

Eg. In PeopleSoft, for batch processes

Either MODULE:

- PSAE.<program name>.<sessionidnum>

Or ACTION:

- PI=<process instance>:…

Join back to application data

- *REGEXP_SUBSTR(h.action,'[[:digit:]]+',4,1)*
- *regexp_substr(h.module,'[^.]+',1,2)*
- *regexp_substr(h.module,'[^.]+',1,3)*

Regexp_replace()

Module defaults to program

- Strip after @ or after .
- *REGEXP_SUBSTR(h.module, '[^.@]+',1,1)*
- *PSAPPSRV@RA48 {TNS V1-V2) → PSAPPSRV*
- *PSAPPSRV.EXE → PSAPPSRV*

numbers to <n>

- *REGEXP_REPLACE(h.action, '[0-9]{2,}','<n>')*
- *DBMS_JOB_12345 → DBMS_JOB_<n>*

BASIC ASH PROFILE BY DB TIME

```
with h as (
select module, action, usecs_per_Row
from dba_hist_Active_Sess_history
where sample_time > sysdate-7
)
select module, action
, sum(usecs_per_row)/1e6 ash_Secs
from h
group by module, action
order by ash_secs desc
fetch first 20 rows only
/
```

```
MODULE                           ACTION                           ASH_SECS
-------------------------------- -------------------------------- ----------
sqldeveloper64W.exe                                               190428.912
CUSTOM_STATS                     PS_LEDGER stats                  150792.593
CUSTOM_STATS                     PS_JRNL_LN stats                  73420.195
SQL*Plus                                                           42311.647
KDMR_IMCO                        KDMR_IMCO Coordinator             37423.301
KTSJ                             KTSJ Slave                        33286.269
                                                                   22573.310
backup incr datafile             0000082 STARTED16                 17905.843
backup incr datafile             0000240 STARTED16                  17857.96
backup incr datafile             0000161 STARTED16                  16470.65
SUMLEDGER_MVREFRESH              PS_XX_SCONSOL_MV stats            16340.025
SUMLEDGER_MVREFRESH              PS_XX_SCONSOL_MV refresh 16       12995.725
PSAE.GL_JP.73937                 GL_JP_UNPOST.DocSeqLn.UpdtLin.S   10897.299
PSAE.GL_JP.85150                 GL_JP_UNPOST.DocSeqLn.UpdtLin.S   10497.779
restore incr datafile            0000655 STARTED40                 9857.0937
backup incr datafile             0000157 STARTED16                  9727.278
```

REPLACE BLANK MODULE WITH PROGRAM

```
with h as (
select NVL(module, program) module
, action, usecs_per_Row
from dba_hist_Active_Sess_history h
where sample_time > sysdate-7
)
select module, action
, sum(usecs_per_row)/1e6 ash_Secs
from h
group by module, action
order by ash_secs desc
fetch first 50 rows only
/
```

```
MODULE                           ACTION                           ASH_SECS
-------------------------------- -------------------------------- ----------
sqldeveloper64W.exe                                               190295.834
CUSTOM_STATS                     PS_LEDGER stats                  150792.593
CUSTOM_STATS                     PS_JRNL_LN stats                  73420.195
SQL*Plus                                                           42311.647
KDMR_IMCO                        KDMR_IMCO Coordinator             37423.301
KTSJ                             KTSJ Slave                        33286.268
backup incr datafile             0000082 STARTED16                 17905.843
backup incr datafile             0000240 STARTED16                 17857.961
backup incr datafile             0000161 STARTED16                 16470.648
SUMLEDGER_MVREFRESH              PS_XX_SCONSOL_MV stats            16340.025
SUMLEDGER_MVREFRESH              PS_XX_SCONSOL_MV refresh 16       12995.725
PSAE.GL_JP.73937                 GL_JP_UNPOST.DocSeqLn.UpdtLin.S   10897.299
PSAE.GL_JP.85150                 GL_JP_UNPOST.DocSeqLn.UpdtLin.S   10497.779
restore incr datafile            0000655 STARTED40                  9857.094
backup incr datafile             0000157 STARTED16                  9727.278
backup incr datafile             0000158 STARTED16                  9582.470
…
```

STRIP .EXE AND @...
PEOPLESOFT SPECIFIC

```
with h as (
select CASE WHEN h.module IS NULL
       THEN REGEXP_SUBSTR(h.program, '[^@]+',1,1)
       WHEN h.module LIKE 'PSAE.%.%'
       THEN REGEXP_SUBSTR(h.module, '[^.]+',1,2)
       ELSE REGEXP_SUBSTR(h.module, '[^.@]+',1,1)
       END AS module, action, usecs_per_Row
from dba_hist_Active_Sess_history h
where sample_time > sysdate-7)
select module, action, sum(usecs_per_row)/1e6 ash_Secs
from h
group by module, action
order by ash_secs desc
fetch first 50 rows only
/
```

```
MODULE                           ACTION                           ASH_SECS
-------------------------------- -------------------------------- ----------
sqldeveloper64W                                                   190029.573
CUSTOM_STATS                     PS_LEDGER stats                  150792.593
GL_JP                            GL_JP_UNPOST.DocSeqLn.UpdtLin.S   78223.472
CUSTOM_STATS                     PS_JRNL_LN stats                  73420.195
SQL*Plus                                                           42311.647
GL_JP                            GL_JP_UNPOST.Process.InsLin.S     42016.400
KDMR_IMCO                        KDMR_IMCO Coordinator             37382.186
KTSJ                             KTSJ Slave                        33265.786
GL_JP                            GL_JP_UNPOST.MAIN.LockJrnl.S      18835.681
backup incr datafile             0000082 STARTED16                 17905.843
backup incr datafile             0000240 STARTED16                 17857.961
GL_JP                            GL_JP_UNPOST.SelHdrs.InsIUTmp.S   16806.647
backup incr datafile             0000161 STARTED16                 16470.648
oracle                                                             16454.713
SUMLEDGER_MVREFRESH              PS_XX_SCONSOL_MV stats            16340.025
…
```

STRIP NUMBERS FROM ORACLE PROCESSES

```
with h as (
select CASE WHEN h.module IS NULL
            THEN REGEXP_SUBSTR(h.program, '[^@]+',1,1)
            WHEN h.module LIKE 'PSAE.%.%'
            THEN REGEXP_SUBSTR(h.module, '[^.]+',1,2)
            ELSE REGEXP_SUBSTR(h.module, '[^.@]+',1,1)
       END AS module
,      CASE WHEN h.module IN('backup incr datafile'
        ,'restore incr datafile','DBMS_SCHEDULER')
            THEN REGEXP_REPLACE(h.action, '[0-9]{2,}','<n>',1,0)
            ELSE action END AS action, usecs_per_Row
from dba_hist_Active_Sess_history h
where sample_time > sysdate-7)
select module, action, sum(usecs_per_row)/1e6 ash_Secs
from h
group by module, action
order by ash_secs desc
fetch first 50 rows only
/
```

```
MODULE                           ACTION                           ASH_SECS
-------------------------------- -------------------------------- ----------
sqldeveloper64W                                                   189660.908
CUSTOM_STATS                     PS_LEDGER stats                  150792.593
backup incr datafile             <n> STARTED<n>                   123601.334
GL_JP                            GL_JP_UNPOST.DocSeqLn.UpdtLin.S   78223.472
CUSTOM_STATS                     PS_JRNL_LN stats                  73420.195
restore incr datafile            <n> STARTED<n>                    62845.079
SQL*Plus                                                           42311.647
GL_JP                            GL_JP_UNPOST.Process.InsLin.S     42016.400
KDMR_IMCO                        KDMR_IMCO Coordinator             37341.228
DBMS_SCHEDULER                   ORA$AT_OS_OPT_SY_<n>              35751.694
KTSJ                             KTSJ Slave                        33265.786
DBMS_SCHEDULER                   ORA$AT_SQ_SQL_SW_<n>              22731.006
GL_JP                            GL_JP_UNPOST.MAIN.LockJrnl.S      18835.681
GL_JP                            GL_JP_UNPOST.SelHdrs.InsIUTmp.S   16806.647
oracle                                                             16464.965
…
```

WHY DOES THIS MATTER?

In the Cloud,
Performance is instrumented as Cost

Either,

- **there are times when you have no free CPU,**
- **use the Resource Manager to prioritise processing.**

Or

- **you are spending too much money on too many OCPUs.**

INSTRUMENTATION CAN BE USED TO DRIVE THE RESOURCE MANAGER

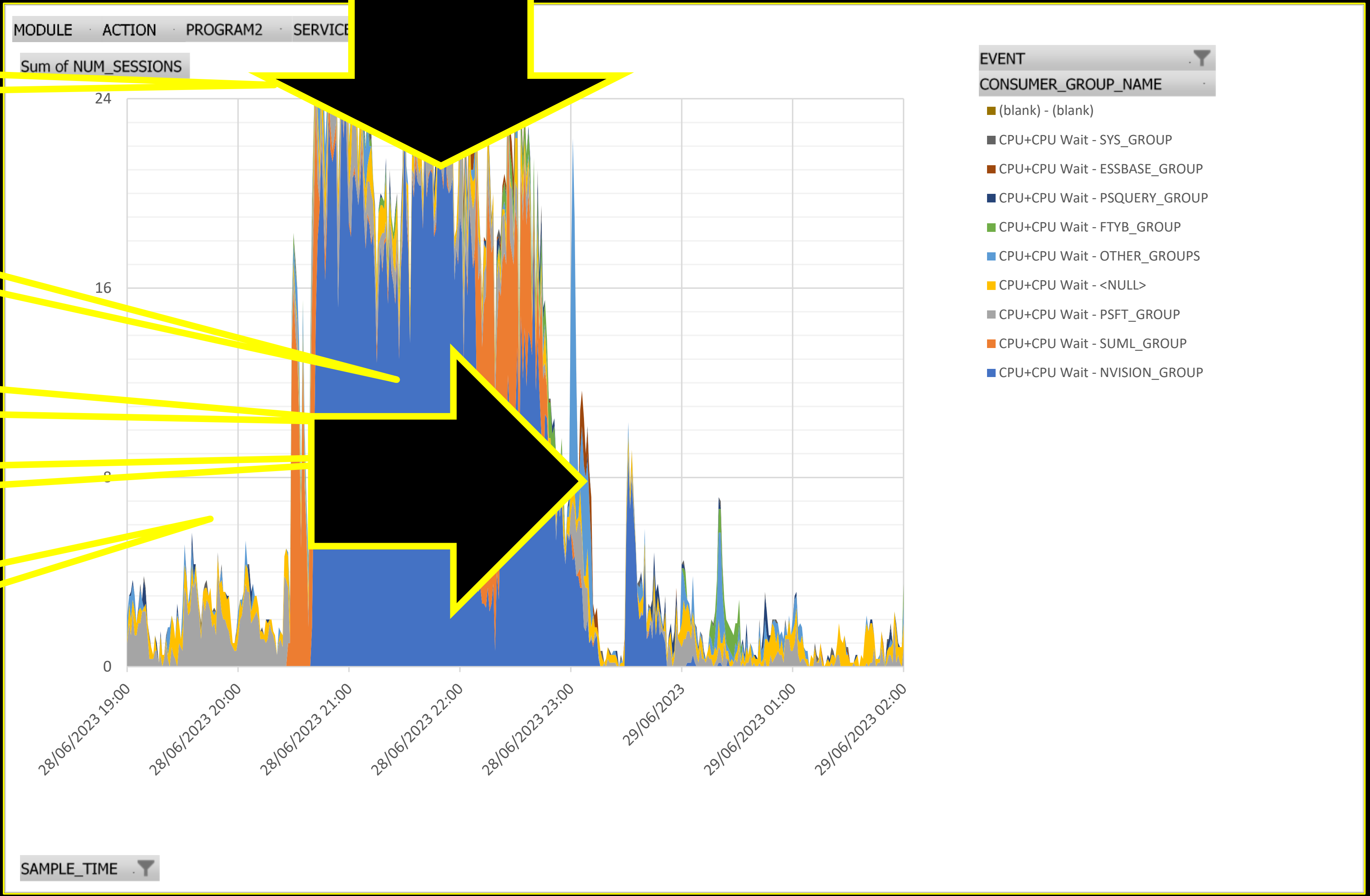
Paying for OCPU

High Priority

Using coloured areas

Lower Priority

Paying for unused white areas



IN THE CLOUD, PERFORMANCE IS INSTRUMENTED AS COST

Instrumentation

- **tells you when and where you are spending your money.**

Resource manager

- **controls how you spend it.**