

Advanced Oracle Troubleshooting

*No magic is needed,
systematic approach will do*

Tanel Põder

<http://www.tanelpoder.com>



Introduction

- About me:
 - Occupation: DBA, researcher, consultant
 - Expertise: Oracle internals geek, *End-to-end* performance & scalability, troubleshooting
 - Oracle experience: 12+ years as DBA
 - Certification: OCM (2002) OCP (1999)
 - Professional affiliations: OakTable Network
 - Blog: <http://blog.tanelpoder.com>



Introduction

- About this presentation:
 - Systematic *approach*, rather than *methodology*
 - Use *right* tools for *right* problems
 - Break complex problems down to simple problems
 - Therefore, use simple tools for simple problems
 - In other words, use a *systematic approach* and life will be easier!
- Less slides, more action!
 - ...this also means I may need to skip some slides
- All scripts used here are freely available:
 - <http://www.tanelpoder.com>

Quiz

Simple (but common) question:

What the \$#*&%! is that session doing?

Non-systematic troubleshooting

- Check alert.log...
- Check for disk and tablespace free space...
- Check for locks...
- Check for xyz...

"We did a healthcheck and everything looks OK!"

? ? ? ? ? !

Semi-systematic troubleshooting

- Quick check for *usual suspects*
 - System load, locks, etc...
- Look into Statspack (or AWR)...
- Enable sql_trace...

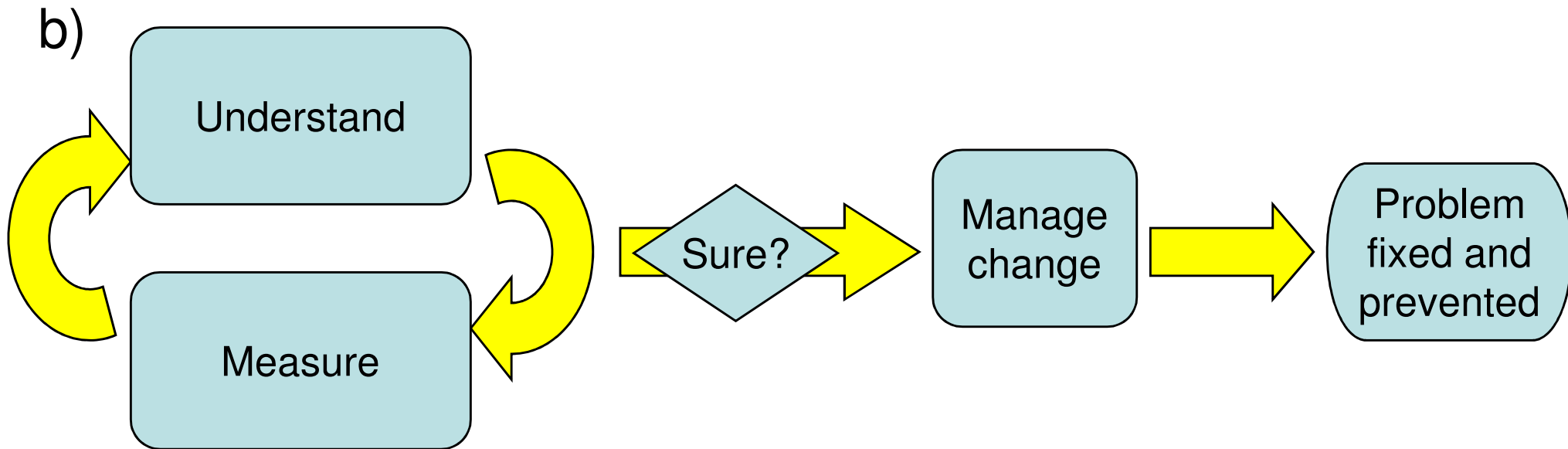
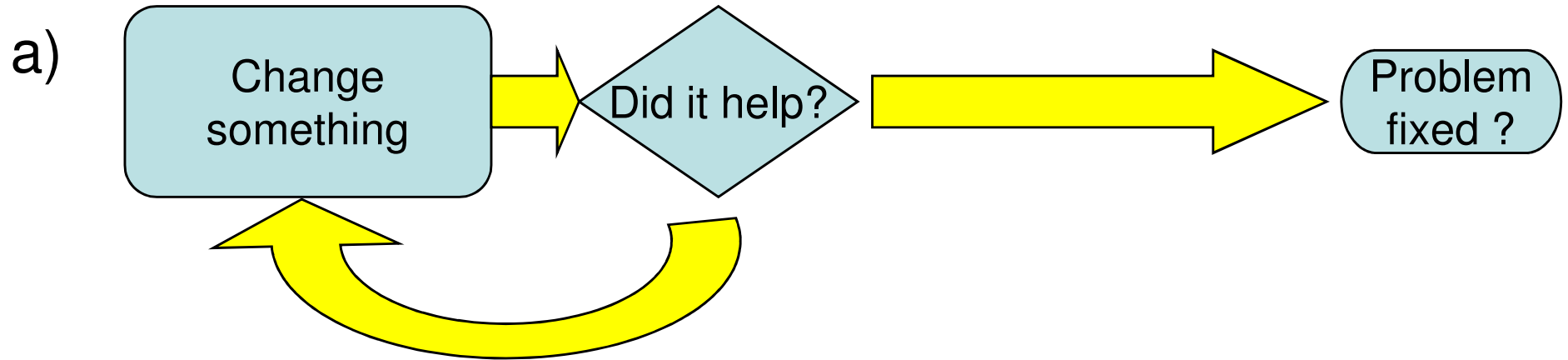
...then what?

Systematic troubleshooting

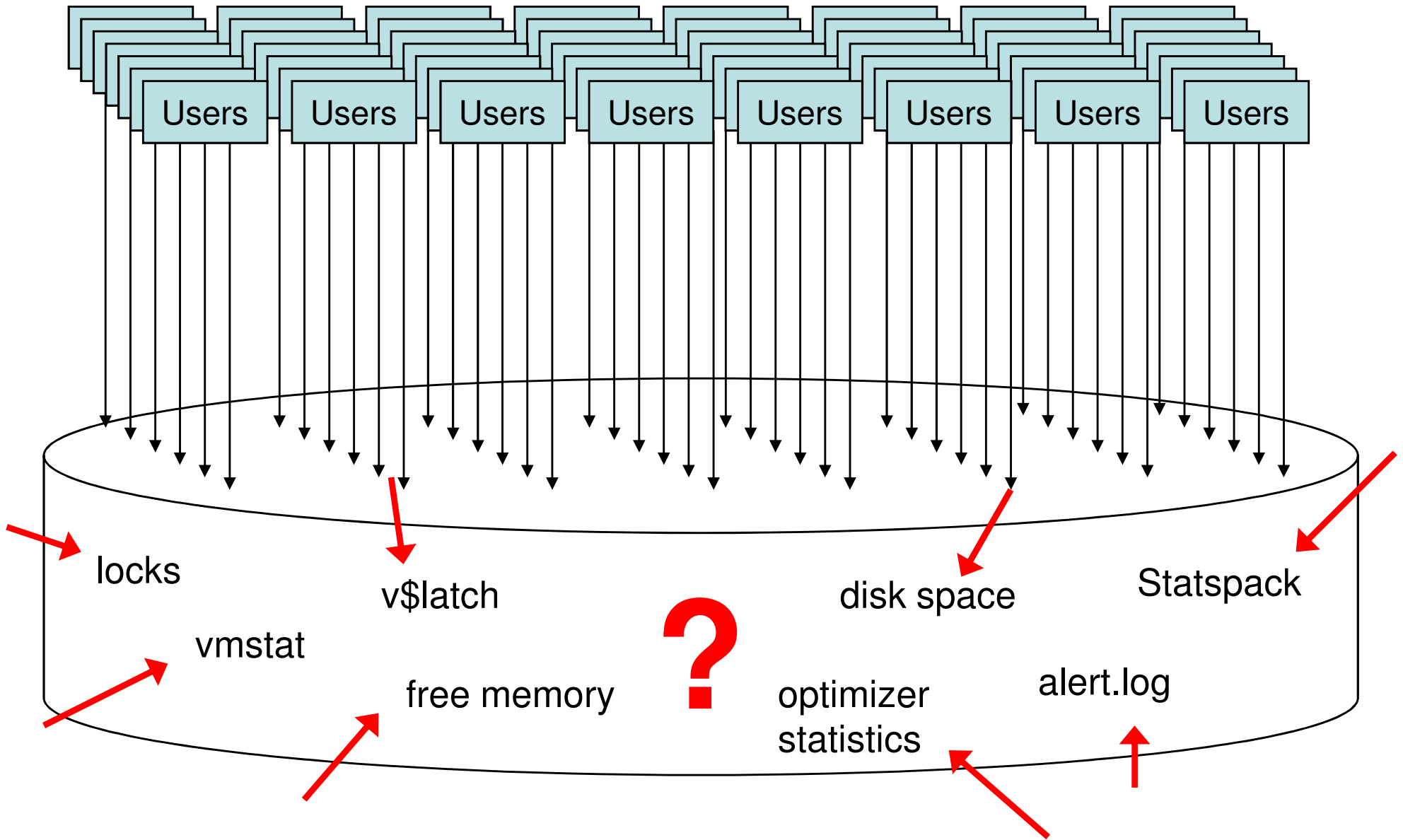
Demo

Troubleshooting approaches

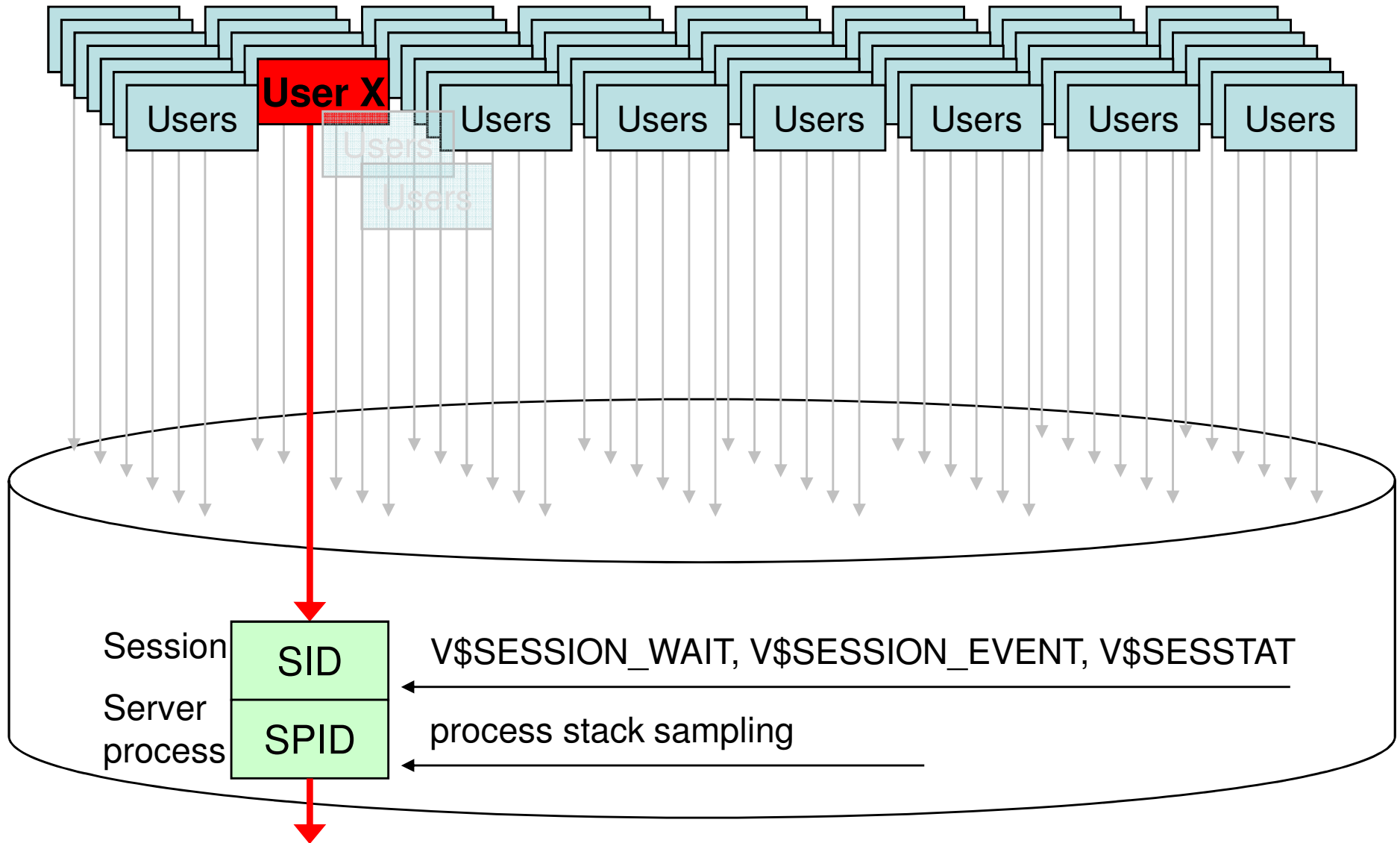
- How do you solve problems?



Non-Systematic approach!

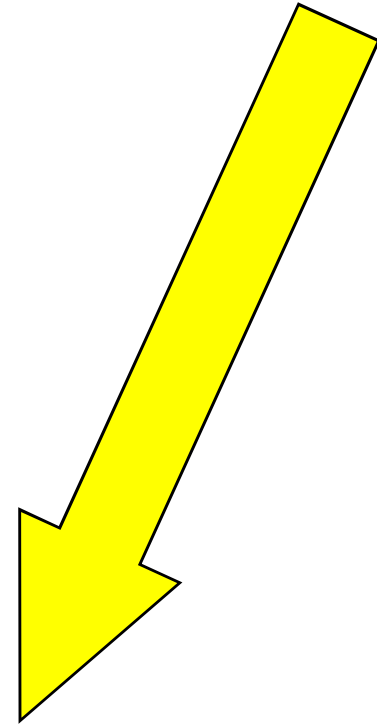


Systematic approach!



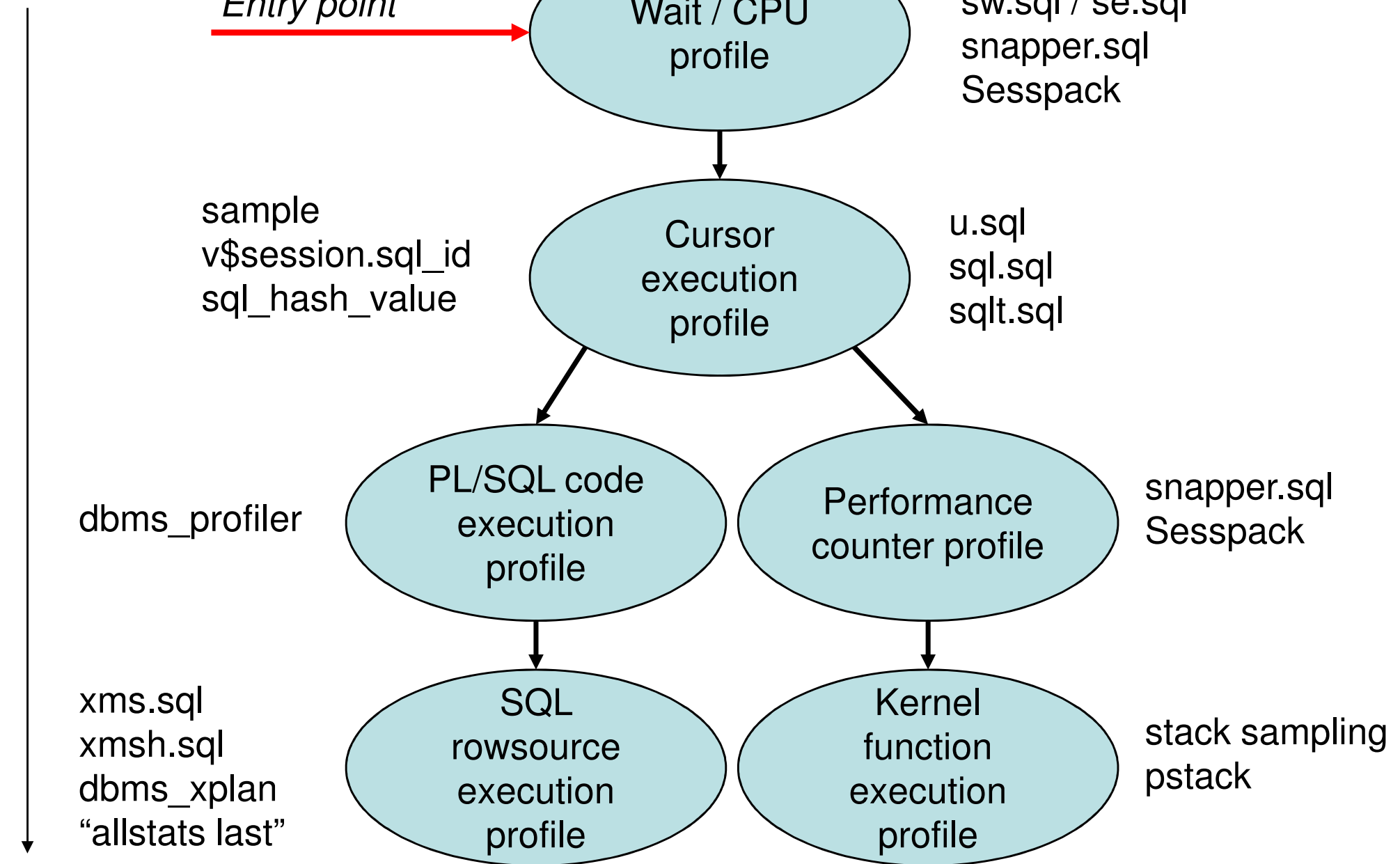
Systematic troubleshooting

- Understand the "flow" of a server process
- ...and how to measure it
- ...then measure it
- ...step by step
- ...using right tool at right step
- ...fix the problem *once you understand it*



Right tools for *measuring* right problems

Detail level



sw.sql and Snapper demo

```
SQL> @sw 114
```

SID	STATE	EVENT	SEQ#	SEC_IN_WAIT	P1	P2	P3	P1TRANSL
114	WAITING	enq: TX - row lock contention	21	9	1415053318	131081	2381	0x54580006: TX mode 6

```
SQL> @sw &mysid
```

SID	STATE	EVENT	SEQ#	SEC_IN_WAIT	P1	P2	P3	P1TRANSL
107	WORKING	On CPU / runqueue	89	0	1413697536	1	0	

```
SQL>
```

```
SQL> @sn 5 &mysid
```

```
-- Session Snapper v1.06 by Tanel Poder ( http://www.tanelpoder.com )
```

HEAD,	SID,	SNAPSHOT	START	SECONDS,	TYPE,	STATISTIC	DELTA,	DELTA/SEC,	HDELTA,	HDELTA/SEC
DATA,	9,	20080221	22:05:08,	5,	STAT,	recursive calls	1,	0,	1,	.2
DATA,	9,	20080221	22:05:08,	5,	STAT,	recursive cpu usage	1,	0,	1,	.2
DATA,	9,	20080221	22:05:08,	5,	STAT,	session pga memory max	25292,	5058,	25.29k,	5.06k
DATA,	9,	20080221	22:05:08,	5,	STAT,	calls to get snapshot scn: kcmgss	1,	0,	1,	.2
DATA,	9,	20080221	22:05:08,	5,	STAT,	workarea executions - optimal	18,	4,	18,	3.6
DATA,	9,	20080221	22:05:08,	5,	STAT,	execute count	1,	0,	1,	.2
DATA,	9,	20080221	22:05:08,	5,	STAT,	sorts (memory)	11,	2,	11,	2.2
DATA,	9,	20080221	22:05:08,	5,	STAT,	sorts (rows)	1904,	381,	1.9k,	380.8
DATA,	9,	20080221	22:05:08,	5,	WAIT,	PL/SQL lock timer	4999649,	999930,	5s,	999.93ms

```
-- End of snap 1
```

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

If this doesn't help...

...Where to look next?

Session troubleshooting sequence

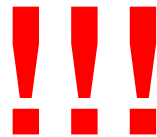
1. Oracle Wait Interface - *response **TIME!***



2. v\$sesstat performance counters - *hints, indicators*



3. Process stack - *truth about process execution*



Process stack demos

```
$ pstack 5855
#0 0x00c29402 in __kernel_vsyscall ()
#1 0x005509e4 in semtimedop () from /lib/libc.so.6
#2 0x0e5769b7 in sskgpwwait ()
#3 0x0e575946 in skgpwwait ()
#4 0x0e2c3adc in ksliwat ()
#5 0x0e2c3449 in kslwaitctx. ()
#6 0x0b007261 in kjusuc ()
#7 0x0c8a7961 in ksipgetctx ()
#8 0x0e2d4dec in ksqcmi ()
#9 0x0e2ce9b8 in ksqgtlctx ()
#10 0x0e2cd214 in ksqqelctx. ()
#11 0x08754afa in ktcwit1 ()
#12 0x0e39b2a8 in kdddgb ()
#13 0x08930c80 in kdddel ()
#14 0x0892af0f in kaudel ()
#15 0x08c3d21a in delrow ()
#16 0x08e6ce16 in qerdlFetch ()
#17 0x08c403c5 in delexe ()
#18 0x0e3c3fa9 in opiexe ()
#19 0x08b54500 in kpoal8 ()
#20 0x0e3be673 in opiodr ()
#21 0x0e53628a in ttcpip ()
#22 0x089a87ab in opitsk ()
#23 0x089aaa00 in opiino ()
#24 0x0e3be673 in opiodr ()
#25 0x089a4e76 in opidrv ()
#26 0x08c1626f in sou2o ()
#27 0x08539aeb in opimai_real ()
#28 0x08c19a42 in ssthrdmain ()
#29 0x08539a68 in main ()
```

Where to look up the meaning of Oracle kernel function names?

1) Metalink:

175982.1 ORA-600 Lookup Error Categories

453521.1 ORA-04031 “KSFQ Buffers” ksmlgpalloc

Search: <function> “Main executable entry point”

2) Oracle views

v\$latch_misses (lm.sql)

v\$latchholder (latchprofx.sql)

v\$fixed_view_definition (d.sql, f.sql)

3) Internet search

Getting stack traces

- OS stack dumper
 - pstack - Solaris, Linux, HP-UX
 - procstack - AIX
 - gdb bt, mdb \$c
 - Procwatcher (Metalink note: 459694.1)
- Windows
 - windbg, procexp - but no symbolic function names in oracle.exe :(
- Oracle internal
 - oradebug short_stack
 - oradebug dump errorstack
 - alter session set events '942 trace name errorstack'

Reading SQL plan execution stack

```
$ ./os_explain pstack.txt
SELECT FETCH:
  SORT: Fetch
    HASH JOIN: Fetch
  * HASH JOIN: Fetch
    * VIEW: Fetch
      NESTED LOOP OUTER: Fetch
        NESTED LOOP OUTER: Fetch
          NESTED LOOP JOIN: Fetch
            HASH JOIN: Fetch
              * VIEW: Fetch
                UNION-ALL: Fetch
                  * VIEW: Fetch
                    UNION-ALL: Fetch
                      * NESTED LOOP OUTER: Fetch
                        NESTED LOOP OUTER: Fetch
                          GRANULE ITERATOR: Fetch
                            INDEX: FetchFastFullScan
                              kdirfrs
```

Scripts for low-level Oracle diagnosis

- dstackprof
 - DTrace based stack profiler
- ostackprof
 - oradebug short_stack based stack profiler
- latchprofx
 - session level latch holder profiling script
 - plain SQL!
 - includes info in which function in Oracle kernel the latch get was invoked

SQL performance troubleshooting (pre 11g)

- Can't do real-time SQL execution monitoring before 11g
 - Use `/*+ gather_plan_statistics */` hint instead
 - ...or set `statistics_level=all`
 - Run the query and wait until it's finished

Get execution statistics of *last* query executed in session:

```
select * from table(  
    dbms_xplan.display_cursor(null, null, 'ALLSTATS LAST')  
);
```

```
select * from table(  
    dbms_xplan.display_cursor(null, null,  
    'ADVANCED +ALLSTATS LAST +MEMSTATS LAST')  
);
```

Real time SQL execution monitoring

- Oracle 11g new feature
 - V\$SQL_MONITOR
 - V\$SQL_PLAN_MONITOR
- MONITOR and NO_MONITOR hints

Get execution statistics of *last/current* query executed in session:

```
SELECT
  DBMS_SQLTUNE.REPORT_SQL_MONITOR (
    session_id=>sys_context('userenv','sid'),
    report_level=>'ALL'
  ) as sql_report
FROM dual;
```

Execution Profile (dbms_sqltune.report_sql_monitor)

SQL> @xp 128

REPORT

SQL Monitoring Report

SQL Text

select /*+ ordered use_nl(b) full(a) full(b) */ count(*) from sys.obj\$ a, sys.obj\$ b where a.name = b.name and r

Global Information

Status : **EXECUTING**
Instance ID : 1
Session ID : 128
SQL ID : lvm188y2gv75n
SQL Execution ID : 16777217
Plan Hash Value : 2119813036
Execution Started : 08/14/2008 18:12:52
First Refresh Time : 08/14/2008 18:13:00
Last Refresh Time : 08/14/2008 18:13:20

SQL Plan Monitoring Details

Id	Operation	Name	Rows (Estim)	Cost	Time Active(s)	Start Active	Starts	Rows (Actual)	Activity (percent)
0	SELECT STATEMENT			16502K			1		
1	SORT AGGREGATE		1				1		
-> 2	COUNT STOPKEY				21	+8	1	3006	
-> 3	NESTED LOOPS		116K	16502K	21	+8	1	3006	
-> 4	TABLE ACCESS FULL	OBJ\$	69996	238	21	+8	1	2925	
-> 5	TABLE ACCESS FULL	OBJ\$	2	236	28	+1	2926	3006	100.00

Execution Profile HTML output (type=>'HTML')

SQL Monitor Report - Windows Internet Explorer

C:\tmp\blah.html

File Edit View Favorites Tools Help

SQL Monitor Report

SQL Text

```
select /*+ ordered use_nl(b) full(a) full(b) */ count(*) from sys.obj$ a, sys.obj$ b where a.name = b.name and rownum <= 1000000000000
```

Global Information: EXECUTING

Instance ID : 1
 Session ID : 111
 SQL ID : 1s4yk8ug0s6g2
 SQL Execution ID : 16777216
 Plan Hash Value : 2119813036
 Execution Started : 08/14/2008 18:51:32
 First Refresh Time : 08/14/2008 18:51:40
 Last Refresh Time : 08/14/2008 18:51:52
 Fetch Calls :

Buffer Gets	IO Count	Database Time	Wait Activity
1651K		19s	

SQL Plan Monitoring Details:

Id	Operation	Name	Estimated Rows	Cost	Active Period (21s)	Starts	Actual Rows	Memory	Temp	CPU Activity	Wait Activity
0	SELECT STATEMENT			16502K		1					
1	SORT AGGREGATE		1			1					
»2	COUNT STOPKEY					1	2101				
»3	NESTED LOOPS		116K	16502K		1	2101				
»4	TABLE ACCESS FULL	OBJ\$	69996	238		1	2065				
»5	TABLE ACCESS FULL	OBJ\$	2	236		2066	2101			100%	

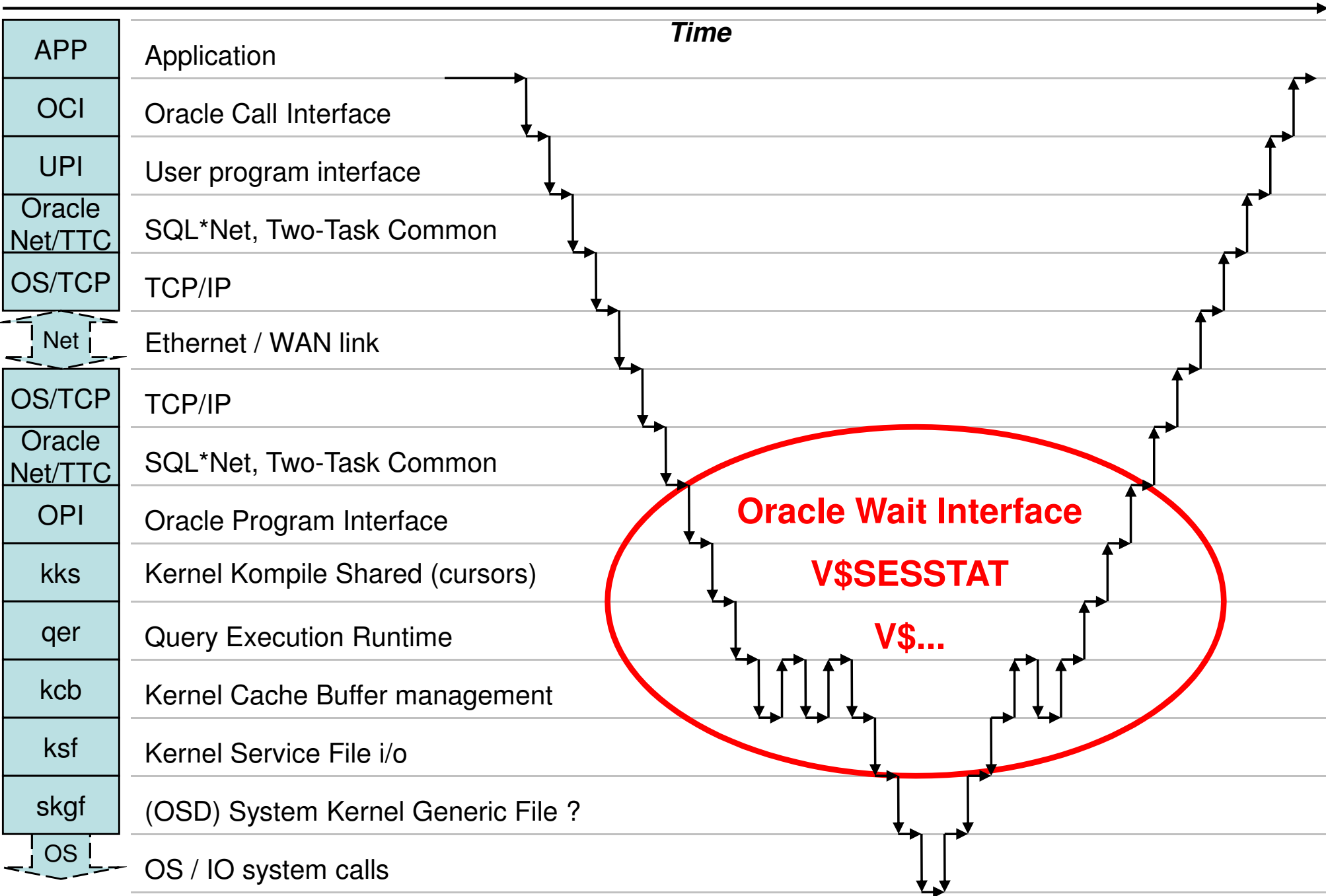
SQL> spool off

My Computer 100%

What if my problem lies outside Oracle?

...Where to look next?

Oracle internal process flow



Oracle internal process flow

APP	Application	Application instrumentation , ltrace, truss -u"libcIntsh:*
OCI	Oracle Call Interface	\$/OH/rdbms/demo/ociucb.mk, OCITrace
UPI	User program interface	-
Oracle Net/TTC	SQL*NET, TNS, Two-Task Common	SQL*Net trace, Wireshark TNS protocol digester
OS/TCP	TCP/IP	Wireshark TCP protocol digester
Net	Ethernet / WAN link	snoop, tcpdump, Wireshark
OS/TCP	TCP/IP	Wireshark TCP protocol digester
Oracle Net/TTC	SQL*NET, TNS, Two-Task Common	SQL*Net trace, Wireshark, Event 10079
OPI	Oracle Program Interface	Event 10051
kks	Kernel Kompile Shared (cursors)	sql_trace, Event 10046, 10270
qer	Query Execution Runtime	v\$sql_plan_statistics, v\$sql_plan_statistics_all, sql_trace
kcb	Kernel Cache Buffer management	x\$kcbsw, Event 10200,10812, _trace_pin_time
ksf	Kernel Service File i/o	v\$filestat, v\$tempstat, v\$session_wait, Event 10298
skgf	(OSD) System Kernel Generic File ?	-
OS	OS / IO system calls	DTrace, strace, truss, tusc, filemon.exe, procmon.exe

Repeat: Session troubleshooting sequence

1. Oracle Wait Interface - *response **TIME!***



2. v\$sesstat performance counters - *hints, indicators*



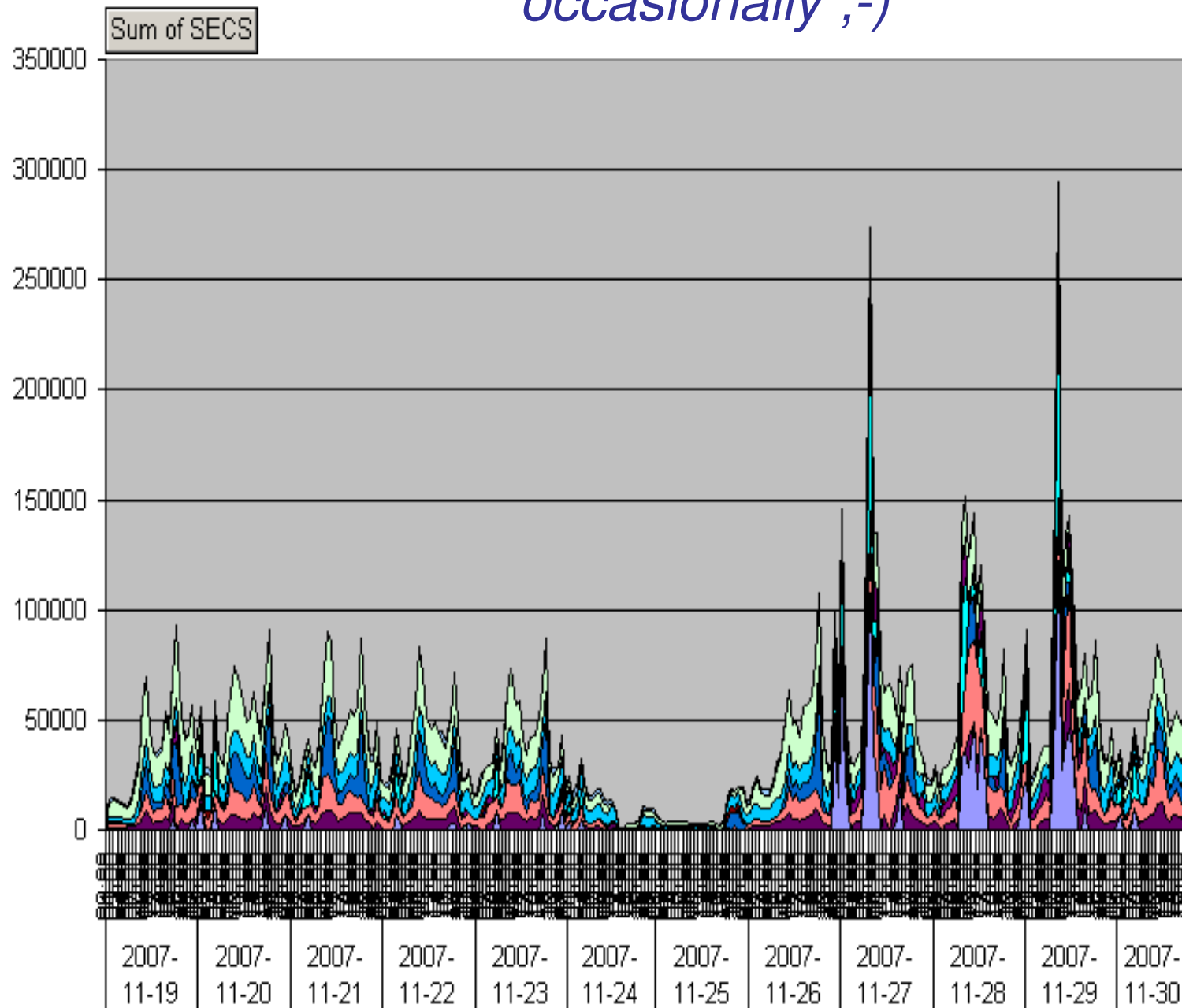
3. Process stack - *truth about process execution*



Session-level troubleshooting tools

1. Profile session wait / CPU breakdown - ***TIME!***
 - *V\$SESSION_WAIT*
 - *V\$SESSION_EVENT, V\$SESS_TIME_MODEL (10g+)*
 - *Snapper, Sesspack, Statspack session mode*
2. Profile session performance counters - *occurrences*
 - *V\$SESSTAT*
3. Profile session's process stack - *statistical sample*
 - *pstack, procstack, ostackprof, dstackprof*
 - *procmon.exe, procexp.exe*
 - *oradebug short_stack, oradebug errorstack*

*A slide to prove that I do use GUIs...
occasionally ;-)*



- NAME
- SQL*Net more data to dblink
 - SQL*Net message from dblink
 - SQL*Net break/reset to client
 - SQL execution
 - single-task message
 - Recursive SQL execution
 - rdbms ipc reply
 - PL/SQL lock timer
 - Parsing SQL
 - log file sync
 - log file switch (checkpoint incomplete)
 - log file sequential read
 - log file parallel write
 - library cache pin
 - library cache load lock
 - latch free
 - db file sequential read
 - db file scattered read
 - db file parallel write
 - db file parallel read
 - control file heartbeat
 - buffer busy waits

DAY HR

Questions?

*Further questions welcome at
<http://blog.tanelpoder.com>*

Thank you!

Tanel Põder

tanel@tanelpoder.com

<http://www.tanelpoder.com>