

A DAY IN THE LIFE OF THE LOG WRITER

FRITS HOOGLAND

ASSUMPTIONS AND WARNING

This talk is about what the logwriter is actually doing, and how to identify contention by understanding. Contention here means "what if the wait event 'log file sync' gets high, indicating LGWR problems"?

The target public is internals specialists, performance architects and troubleshooters.

This talk has nothing to do with daily, operational tasks.

I assume the audience has a reasonable knowledge of:

- Oracle internals in general.**
- C concepts and a sound understanding of function calling and returning.**
 - Understanding what a stack trace is.**
- Internals tools like gdb, pstack, Intel Pin Tools, etc.**

If not, please get out, it will bore you to death; if you want to learn please stay!

VERSIONS

This is what I have used for the investigations:

Operating system: Oracle Linux 7.4 (a vagrant box)
Kernel: 4.1.12-103.9.7.el7uek.x86_64
Oracle database: 12.2.0.1.170814

This is about the internal implementation of the log writer functionality. Anything can change without any notice from Oracle even with a PSU (patch) update.

Different operating systems might (will) have different implementations in certain parts, especially in a layer group called 'VOS' (virtual operating system), like s* (which means 'operating system dependent' AFAIK, and ss* (~ 'system', which often call a platform specific system call (system call wrapper)).

TOOLS

Oracle execution visibility layers:

- SQL call / SQL trace / ASH
- wait events / session counters
- Operating system counters / iostat, sar, etc.
- System call / strace
- Function call / pin tools -> debugtrace
- Memory access patterns / pin tools -> pinatrace

Final call: this deep diving into C execution of a binary called 'oracle'. This is detailed information that is of use to people with an understanding of these layers.

-> As an example of what this is about: there isn't a single SQL statement in this presentation.

WHAT IS THIS TALK ABOUT

In this talk, a microscopic look is taken at what the logwriter is doing.

The database in the investigation is set to specific basic settings that might be different in your database:

- Version 12.2.0.1.170814
- Filesystem based IO
- NOARCHIVELOG
- No RAC
- No DG
- Log writer set not to use worker processes (`_use_single_log_writer=true`)
- No private redo strands in use

I believe this knowledge will set a strong fundament for understanding what the logwriter is doing in all recent versions and across operating systems.

WHAT DOES THE LOG WRITER DO - IDLE?

An idle log writer mostly sleeps on a semaphore:

```
$ strace -p $( pgrep lgwr )
```

```
strace: Process 4670 attached
```

```
getrusage(0x1 /* RUSAGE_??? */, {ru_utime={0, 25232}, ru_stime={0, 7535}, ...}) = 0
```

```
getrusage(0x1 /* RUSAGE_??? */, {ru_utime={0, 25232}, ru_stime={0, 7734}, ...}) = 0
```

```
semtimedop(917504, [{23, -1, 0}], 1, {3, 0}) = -1 EAGAIN (Resource temporarily unavaila
```

```
getrusage(0x1 /* RUSAGE_??? */, {ru_utime={0, 25286}, ru_stime={0, 7780}, ...}) = 0
```

```
getrusage(0x1 /* RUSAGE_??? */, {ru_utime={0, 25295}, ru_stime={0, 7783}, ...}) = 0
```

```
semtimedop(917504, [{23, -1, 0}], 1, {2, 999999000}) = -1 EAGAIN (Resource temporarily
```

```
getrusage(0x1 /* RUSAGE_??? */, {ru_utime={0, 25347}, ru_stime={0, 7799}, ...}) = 0
```

```
getrusage(0x1 /* RUSAGE_??? */, {ru_utime={0, 25353}, ru_stime={0, 7801}, ...}) = 0
```

```
semtimedop(917504, [{23, -1, 0}], 1, {2, 999999000}) = -1 EAGAIN (Resource temporarily
```

accenture[>][operations](#)

KSARCV

```
$ pstack $( pgrep lgwr )
#0  0x00007f7402e15bda in semtimedop () at ../sysdeps/unix/syscall-template.S:81
#1  0x0000000010f9cca6 in sskgpwait () o/s dependent system kernel generic post/wait
#2  0x0000000010f9a2e8 in skgpwait () o/s dependent kernel generic post/wait wait
#3  0x0000000010a66995 in ksliwat () kernel service locking inner wait function;
#4  0x0000000010a65d25 in kslwaitctx () kernel service locking wait until posted/tim
#5  0x0000000010e41ea3 in ksarcv () kernel service async receive message
#6  0x000000000335c874 in ksbabs () kernel service backgrnd procs action based s
#7  0x000000000338a807 in ksbrdp () kernel service backgrnd procs run detached p
#8  0x000000000370d9a1 in opirip () oracle program interface run independent pro
#9  0x0000000001eb034a in opidrv () oracle program interface route current requ
#10 0x0000000002afedf1 in sou2o () main oracle executable entry point
#11 0x0000000000d0547a in opimai_real () oracle program interface main
#12 0x0000000002b09b31 in ssthrdmain () o/s dependent system main for every thread
#13 0x0000000000d05386 in main () general c starting function
```

DEBUGTRACE

A large part of this presentation is spend on looking at Intel PinTools filtered debugtrace output.

A filtered debugtrace shows the C function call flow, like this:

```
$ less lgwr_full_cycle_stripped.txt
<!semtimedop+0x000000000023 returns: 0xffffffffffffffff
> __errno_location(0x38000, 0x7ffc3dd113a8, ...)
| > fthread_self(0x38000, 0x7ffc3dd113a8, ...)
| < fthread_self+0x000000000024 returns: 0
< __errno_location+0x000000000010 returns: 0x7f2c23b9e6a0
<!sskgpwait+0x000000000014e returns: 0
<!skgpwait+0x0000000000e0 returns: 0
> ksuSdiInProgress(0x13a2a, 0x13a2a, ...)
< ksuSdiInProgress+0x000000000035 returns: 0
> sltrgftime64(0x13a2a, 0x13a2a, ...)
| > clock_gettime@plt(0x1, 0x7ffc3dd11420, ...)
| | > clock_gettime(0x1, 0x7ffc3dd11420, ...)
| | < clock_gettime+0x000000000069 returns: 0
```


In order to learn more about the log writer, first it must be established how it performs its tasks. In order to do that, I performed a pin tools debugtrace on the log writer, and only listed the first call level

```
$ awk '/^\ \x3e\ /' lgwr_full_cycle_stripped.txt
> ksarcv(0x7ffc3dd12338, 0x7ffc3dd122a0, ...) <<<<
> ksl_exit_main_loop_wait(0x9d64fb00c, 0x7, ...)
> ksbcti(0x12b84c18, 0x7f2c1eb804b0, ...) (nr 1)
> dbktFlush(0, 0, ...)
> sltrgatime64(0x7f2c23b9e9a0, 0x7f2c23b9ebe8, ...)
> ksbcti(0x12b84be0, 0x7f2c1eb80208, ...) (nr 2)
> ksbcti(0x12b84bfc, 0x7f2c1eb804b0, ...) (nr 3)
> sltrgatime64(0, 0, ...)
> ksl_enter_main_loop_wait(0x1, 0x4ccef9975, ...)
> ksarcv(0x7ffc3dd12338, 0x7ffc3dd122a0, ...) <<<<
> ksl_exit_main_loop_wait(0x9d67d8b01, 0x7, ...)
> ksbcti(0x12b84c18, 0x7f2c1eb804b0, ...) (nr 1)
> dbktFlush(0, 0, ...)
```

KSARCV

```
> ksarcv(0x7ffc3dd12338, 0x7ffc3dd122a0, ...) kernel service async receive message
| > kslgetl(0x6003ef40, 0x1, ...) kernel service locking get latch
| > kslfre(0x6003ef40, 0x1, ...) kernel service locking free latch
| > kslwtbctx(0x7ffc3dd119d0, 0x1, ...) kernel service locking wait begin context
| > kslwaitctx(0x7ffc3dd119d0, 0xef, ...) kernel service locking wait context
| > kslgetl(0x6003ef40, 0x1, ...)
| > kslfre(0x6003ef40, 0x1, ...)
| > kslwtctx(0x7ffc3dd119d0, 0x1, ...) kernel service locking wait end context
```

```
SQL> select addr, child#, name from v$latch
       where to_number(addr, 'XXXXXXXXXXXXXXXXXX') = to_number('6003ef40', 'XXXXXXXXXX');
ADDR          NAME
-----
000000006003EF40 messages
```

WHAT DOES THE LOG WRITER DO?

```
$ awk '/^\ \x3e\ /' lgwr_full_cycle_stripped.txt
> ksarcv(0x7ffc3dd12338, 0x7ffc3dd122a0, ...) <<<<
> ksl_exit_main_loop_wait(0x9d64fb00c, 0x7, ...)
> ksbcti(0x12b84c18, 0x7f2c1eb804b0, ...) (nr 1)
> dbktFlush(0, 0, ...)
> sltrgatime64(0x7f2c23b9e9a0, 0x7f2c23b9ebe8, ...)
> ksbcti(0x12b84be0, 0x7f2c1eb80208, ...) (nr 2)
> ksbcti(0x12b84bfc, 0x7f2c1eb804b0, ...) (nr 3)
> sltrgatime64(0, 0, ...)
> ksl_enter_main_loop_wait(0x1, 0x4ccef9975, ...)
> ksarcv(0x7ffc3dd12338, 0x7ffc3dd122a0, ...) <<<<
> ksl_exit_main_loop_wait(0x9d67d8b01, 0x7, ...)
> ksbcti(0x12b84c18, 0x7f2c1eb804b0, ...) (nr 1)
> dbktFlush(0, 0, ...)
> sltrgatime64(0x7f2c23b9e9a0, 0x7f2c23b9ebe8, ...)
> ksbcti(0x12b84be0, 0x7f2c1eb80208, ...) (nr 2)
> ksbcti(0x12b84bfc, 0x7f2c1eb804b0, ...) (nr 3)
```

DBKTFLUSH

```
> dbktFlush(0, 0, ...)                                rdbms UTS tracefile - Flush
| > dbgtfdFileUseAdr(0x7f2c23b5e6c0, 0, ...)           diagnostic component test parent dir
| < dbgtfdFileUseAdr+0x00000000000040 returns: 0x1
| > dbgtfFlush(0x7f2c23b5e6c0, 0x7f2c23b6a088, ...)
| | > dbgtfGetActiveFile(0x7f2c23b5e6c0, 0x7ffc3dd119f0, ...)
| | < dbgtfGetActiveFile+0x00000000000054 returns: 0x1
| | > __setjmp@plt(0x7ffc3dd118e8, 0x7ffc3dd119b8, ...)
| | <> __sigsetjmp(0x7ffc3dd118e8, 0, ...)
| | <> __sigjmp_save(0x7ffc3dd118e8, 0, ...)
| | < __sigjmp_save+0x00000000000025 returns: 0
| | > dbgtfdFileAccessCbk(0x7f2c23b5e6c0, 0x7f2c23b6a088, ...)
| | < dbgtfdFileAccessCbk+0x000000000000199 returns: 0x1
| | > dbgtfdFileAccessCbk(0x7f2c23b5e6c0, 0x7f2c23b6a088, ...)
| | < dbgtfdFileAccessCbk+0x000000000000199 returns: 0x1
| < dbgtfFlush+0x000000000000274 returns: 0x1
< dbktFlush+0x00000000000006b returns: 0x1
```

WHAT DOES THE LOG WRITER DO?

```
$ awk -n '$0 ~ /^\\ \x3e\ /\ ' lgwr_full_cycle_stripped.txt
> ksarcv(0x7ffc3dd12338, 0x7ffc3dd122a0, ...) <<<<
> ksl_exit_main_loop_wait(0x9d64fb00c, 0x7, ...)
> ksbcti(0x12b84c18, 0x7f2c1eb804b0, ...) (nr 1)
> dbktFlush(0, 0, ...)
> sltrgatime64(0x7f2c23b9e9a0, 0x7f2c23b9ebe8, ...)
> ksbcti(0x12b84be0, 0x7f2c1eb80208, ...) (nr 2)
> ksbcti(0x12b84bfc, 0x7f2c1eb804b0, ...) (nr 3)
> sltrgatime64(0, 0, ...)
> ksl_enter_main_loop_wait(0x1, 0x4ccef9975, ...)
> ksarcv(0x7ffc3dd12338, 0x7ffc3dd122a0, ...) <<<<
> ksl_exit_main_loop_wait(0x9d67d8b01, 0x7, ...)
> ksbcti(0x12b84c18, 0x7f2c1eb804b0, ...) (nr 1)
> dbktFlush(0, 0, ...)
> sltrgatime64(0x7f2c23b9e9a0, 0x7f2c23b9ebe8, ...)
> ksbcti(0x12b84be0, 0x7f2c1eb80208, ...) (nr 2)
> ksbcti(0x12b84bfc, 0x7f2c1eb804b0, ...) (nr 3)
```

SLTRGATIME64

```
> sltrgtime64(0x7f2c23b9e9a0, 0x7f2c23b9ebe8, ...) o/s dependent call arbitrary time
| > sltrgftime64(0x7f2c23b9e9a0, 0x7f2c23b9ebe8, ...) o/s dependent call time lookup ma
| | > clock_gettime@plt(0x1, 0x7ffc3dd11b60, ...)
| | | > clock_gettime(0x1, 0x7ffc3dd11b60, ...)
| | | < clock_gettime+0x00000000000069 returns: 0
| | < clock_gettime+0x0000000000003a returns: 0
| < sltrgftime64+0x0000000000004c returns: 0x9cb90dc15
< sltrgtime64+0x0000000000003e returns: 0x281efd6
```

WHAT DOES THE LOG WRITER DO?

```
$ awk -n '$0 ~ /^\\ \x3e\ /\ ' lgwr_full_cycle_stripped.txt
> ksarcv(0x7ffc3dd12338, 0x7ffc3dd122a0, ...) <<<<
> ksl_exit_main_loop_wait(0x9d64fb00c, 0x7, ...)
> ksbcti(0x12b84c18, 0x7f2c1eb804b0, ...) (nr 1)
> dbktFlush(0, 0, ...)
> sltrgatime64(0x7f2c23b9e9a0, 0x7f2c23b9ebe8, ...)
> ksbcti(0x12b84be0, 0x7f2c1eb80208, ...) (nr 2)
> ksbcti(0x12b84bfc, 0x7f2c1eb804b0, ...) (nr 3)
> sltrgatime64(0, 0, ...)
> ksl_enter_main_loop_wait(0x1, 0x4ccef9975, ...)
> ksarcv(0x7ffc3dd12338, 0x7ffc3dd122a0, ...) <<<<
> ksl_exit_main_loop_wait(0x9d67d8b01, 0x7, ...)
> ksbcti(0x12b84c18, 0x7f2c1eb804b0, ...) (nr 1)
> dbktFlush(0, 0, ...)
> sltrgatime64(0x7f2c23b9e9a0, 0x7f2c23b9ebe8, ...)
> ksbcti(0x12b84be0, 0x7f2c1eb80208, ...) (nr 2)
> ksbcti(0x12b84bfc, 0x7f2c1eb804b0, ...) (nr 3)
```

KSBCTI - 1

kernel service background processes call timeout/interrupts

```
$ awk '/^\ \x3e\ ksbcti\ (0x12b84c18/,/^\ \x3c\ ksbcti/' lgwr_full_cycle_stripped.txt\  
| awk '/^\ \x3e\ / || /^\ \|\ \x3e\ /'
```

```
> ksbcti(0x12b84c18, 0x7f2c1eb804b0, ...)  
| > ksbmsg(0, 0, ...) kernel service bg cpu count changed callbacks  
| > ksu_dispatch_tac(0, 0, ...) kernel service user dispatch timed-out actions  
| > ksbxiaf(0, 0, ...) kernel service bg ksbxic message and interrupt actions  
| > kjci_action(0, 0, ...) kernel lock management communication cross instance  
| > kcfsmpoll(0, 0, ...) kernel cache file management ??  
| > ksuwaitsysevent(0, 0, ...) kernel service user deal with channels (x$ksrchdl, x$ksrchkd)  
| > krdrsb_lgwr_chintr(0, 0, ...) kernel recovery datafiles read from standby ??  
| > ksbckbast(0, 0, ...) kernel service bg check blocking asynchronous trap
```


KSBCTI - 1 - KSU_DISPATCH_TAC

```
| > ksu_dispatch_tac(0, 0, ...)
| | > _setjmp@plt(0x7ffc3dd10c50, 0, ...)
| | <> __sigsetjmp(0x7ffc3dd10c50, 0, ...)
| | <> __sigjmp_save(0x7ffc3dd10c50, 0, ...)
| | < __sigjmp_save+0x0000000000025 returns: 0
| | > dbkc_diag_tac_cb(0x7f2c23b89410, 0, ...)
| | | > _setjmp@plt(0x7ffc3dd10a40, 0, ...)
| | | <> __sigsetjmp(0x7ffc3dd10a40, 0, ...)
| | | <> __sigjmp_save(0x7ffc3dd10a40, 0, ...)
| | | < __sigjmp_save+0x0000000000025 returns: 0
| | | > ksu_deactivate_tac(0x7f2c23b89410, 0, ...)
| | | < ksu_deactivate_tac+0x000000000000e returns: 0
| | | > dbkdSyncSysTac(0, 0, ...)
| | | < dbkdSyncSysTac+0x0000000000278 returns: 0x75efe388
| | | > kspgip(0x1235, 0x1, ...)
| | | < kspgip+0x000000000023f returns: 0
| | | > kspgip(0x1235, 0x1, ...)
| | | < kspgip+0x000000000023f returns: 0
| | | > dbgtfdAutoClose(0x7f2c23b5e6c0, 0x5a4fe5b7, ...)
| | | | > dbgtfdFileGetDefault(0x7f2c23b5e6c0, 0x5a4fe5b7, ...)
```

```
| | | <> __sigsetjmp(0x7ffc3dd10a40, 0, ...)
| | | <> __sigjmp_save(0x7ffc3dd10a40, 0, ...)
| | | < __sigjmp_save+0x0000000000025 returns: 0
| | | > ksu_deactivate_tac(0x7f2c23b89410, 0, ...)
| | | < ksu_deactivate_tac+0x000000000000e returns: 0
| | | > dbkdSyncSysTac(0, 0, ...)
| | | < dbkdSyncSysTac+0x0000000000278 returns: 0x75efe388
| | | > kspgip(0x1235, 0x1, ...)
| | | < kspgip+0x000000000023f returns: 0
| | | > kspgip(0x1235, 0x1, ...)
| | | < kspgip+0x000000000023f returns: 0
| | | > dbgtdfdAutoClose(0x7f2c23b5e6c0, 0x5a4fe5b7, ...)
| | | | > dbgtdfdFileGetDefault(0x7f2c23b5e6c0, 0x5a4fe5b7, ...)
| | | | < dbgtdfdFileGetDefault+0x0000000000027 returns: 0x7f2c23b6a088
| | | < dbgtdfdAutoClose+0x00000000000a8 returns: 0xfa0
| | | > dbkrlSyncSysTac(0, 0x5a4fe5b7, ...)
| | | < dbkrlSyncSysTac+0x000000000002f returns: 0xfa0
| | | > ksu_unmask_tac(0x7f2c23b89410, 0x7f2c23b9ebe8, ...)
| | | < ksu_unmask_tac+0x000000000002b returns: 0x5a4fe5b5
| | < dbkc_diag_tac_cb+0x00000000002b1 returns: 0
| | > _setjmp@plt(0x7ffc3dd10c50, 0, ...)
| | <> __sigsetjmp(0x7ffc3dd10c50, 0, ...)
| | <> __sigjmp_save(0x7ffc3dd10c50, 0, ...)
| | < __sigjmp_save+0x0000000000025 returns: 0
```

KSBCTI - 1 - KSU_DISPATCH_TAC

kews_timeout 1/2

```
| | > kews_timeout(0x7f2c1eb93ec0, 0, ...)          kernel event AWR statistics timeout
| | | > sltrgftime64(0x7f2c1eb93ec0, 0, ...)      o/s dependent call time lookup main
| | | | > clock_gettime@plt(0x1, 0x7ffc3dd10ad0, ...)
| | | | | > clock_gettime(0x1, 0x7ffc3dd10ad0, ...)
| | | | | < clock_gettime+0x00000000000069 returns: 0
| | | | | < clock_gettime+0x0000000000003a returns: 0
| | | | < sltrgftime64+0x0000000000004c returns: 0x9cb6163fa
| | | | > slcpums(0x7f2c23ba3c58, 0x191ae1d5, ...) o/s dependent call get cpu usage mea
| | | | | > getrusage(0x1, 0x7ffc3dd10a30, ...)
| | | | | | > fthread_self(0x1, 0x7ffc3dd10a30, ...)
| | | | | | < fthread_self+0x00000000000024 returns: 0
| | | | | <> getrusage_ext(0x1, 0x7ffc3dd10a30, ...)
| | | | | < getrusage+0x0000000000000f returns: 0
| | | | < slcpums+0x0000000000008f returns: 0x31d592
```

KSBCTI - 1 - KSU_DISPATCH_TAC

kews_timeout 2/2

```
| | | > kews_sqlst_flush(0xc, 0x72fd7d40, ...) kernel event AWR statistics flush curs
| | | | > sltrgftime64(0xc, 0x72fd7d40, ...) o/s dependent call time lookup main (t
| | | | | > clock_gettime@plt(0x1, 0x7ffc3dd10a60, ...)
| | | | | | > clock_gettime(0x1, 0x7ffc3dd10a60, ...)
| | | | | | < clock_gettime+0x00000000000069 returns: 0
| | | | | | < clock_gettime+0x0000000000003a returns: 0
| | | | | < sltrgftime64+0x0000000000004c returns: 0x9cb61723f
| | | | | > slcpums(0x7f2c23ba3c58, 0x7ae0ee70, ...) o/s dependent call get cpu usage meas
| | | | | | > getrusage(0x1, 0x7ffc3dd109c0, ...)
| | | | | | | > fthread_self(0x1, 0x7ffc3dd109c0, ...)
| | | | | | | < fthread_self+0x00000000000024 returns: 0
| | | | | | <> getrusage_ext(0x1, 0x7ffc3dd109c0, ...)
| | | | | | < getrusage+0x0000000000000f returns: 0
| | | | | < slcpums+0x0000000000008f returns: 0x31e417
| | | | < kews_sqlst_flush+0x0000000000016a returns: 0x7f2c23b9e900
| | | < kews_timeout+0x0000000000052c returns: 0x7f2c23b9e900
```

WHAT DOES THE LOG WRITER DO?

```
$ awk -n '$0 ~ /^\\ \x3e\ /\ ' lgwr_full_cycle_stripped.txt
> ksarcv(0x7ffc3dd12338, 0x7ffc3dd122a0, ...) <<<<
> ksl_exit_main_loop_wait(0x9d64fb00c, 0x7, ...)
> ksbcti(0x12b84c18, 0x7f2c1eb804b0, ...) (nr 1)
> dbktFlush(0, 0, ...)
> sltrgatime64(0x7f2c23b9e9a0, 0x7f2c23b9ebe8, ...)
> ksbcti(0x12b84be0, 0x7f2c1eb80208, ...) (nr 2)
> ksbcti(0x12b84bfc, 0x7f2c1eb804b0, ...) (nr 3)
> sltrgatime64(0, 0, ...)
> ksl_enter_main_loop_wait(0x1, 0x4cce9975, ...)
> ksarcv(0x7ffc3dd12338, 0x7ffc3dd122a0, ...) <<<<
> ksl_exit_main_loop_wait(0x9d67d8b01, 0x7, ...)
> ksbcti(0x12b84c18, 0x7f2c1eb804b0, ...) (nr 1)
> dbktFlush(0, 0, ...)
> sltrgatime64(0x7f2c23b9e9a0, 0x7f2c23b9ebe8, ...)
> ksbcti(0x12b84be0, 0x7f2c1eb80208, ...) (nr 2)
> ksbcti(0x12b84bfc, 0x7f2c1eb804b0, ...) (nr 3)
```


WHAT DOES THE LOG WRITER DO?

```
$ awk '/^\ \x3e\ /' lgwr_full_cycle_stripped.txt
> ksarcv(0x7ffc3dd12338, 0x7ffc3dd122a0, ...) <<<<
> ksl_exit_main_loop_wait(0x9d64fb00c, 0x7, ...)
> ksbcti(0x12b84c18, 0x7f2c1eb804b0, ...) (nr 1)
> dbktFlush(0, 0, ...)
> sltrgatime64(0x7f2c23b9e9a0, 0x7f2c23b9ebe8, ...)
> ksbcti(0x12b84be0, 0x7f2c1eb80208, ...) (nr 2)
> ksbcti(0x12b84bfc, 0x7f2c1eb804b0, ...) (nr 3)
> sltrgatime64(0, 0, ...)
> ksl_enter_main_loop_wait(0x1, 0x4ccef9975, ...)
> ksarcv(0x7ffc3dd12338, 0x7ffc3dd122a0, ...) <<<<
> ksl_exit_main_loop_wait(0x9d67d8b01, 0x7, ...)
> ksbcti(0x12b84c18, 0x7f2c1eb804b0, ...) (nr 1)
> dbktFlush(0, 0, ...)
> sltrgatime64(0x7f2c23b9e9a0, 0x7f2c23b9ebe8, ...)
> ksbcti(0x12b84be0, 0x7f2c1eb80208, ...) (nr 2)
> ksbcti(0x12b84bfc, 0x7f2c1eb804b0, ...) (nr 3)
```

KSBCTI - 2

```
$ awk '/^\ \x3e\ ksbcti\(0x12b84be0/, /^\ \x3c\ ksbcti/' lgwr_full_cycle_stripped.txt\  
| awk '/^\ \x3e\ / || /^\ \|\ \x3e\ /'
```

```
> ksbcti(0x12b84be0, 0x7f2c1eb80208, ...)  
| > ksbckbast(0, 0, ...) kernel service bg check blocking asynchronous  
| > ksumcl(0, 0, ...) kernel service user multi check for proc all  
| > kcrfw_redo_write_driver(0, 0, ...) kernel cache redo file write main redo write  
| > kcrfwl(0, 0, ...) kernel cache redo file switch logs  
| > kcrfwg(0, 0, ...) kernel cache redo file write ??  
| > kcrf_cache_all_logs(0, 0, ...) kernel cache redo file ??  
| > krd_rcvinf_cleanup_callback(0, 0, ...) kernel recovery datafiles ??  
| > kfrbDriver(0, 0, ...) kernel ASM recovery ??  
| > kfrbCkpt(0, 0, ...) kernel ASM recovery ??  
| > kfrbRelocateACT(0, 0, ...) kernel ASM recovery ??  
| > ksbckbast(0, 0, ...) kernel service bg check blocking asynchronous
```


KCRFW_REDO_WRITE_DRIVER

```
awk '/^\ \|\ \x3e\ kcrfw_redo_write_driver/,/^\ \|\ \x3c\ kcrfw_redo_write_driver/'  
lgwr_full_cycle_stripped.txt | awk '/^\ \|\ \x3e\ /|/^\ \|\ \|\ \x3e/'
```

```
| > kcrfw_redo_write_driver(0, 0, ...) kernel cache redo file writ  
| | > kcrfw_handle_member_write_errors(0, 0, ...) kernel cache redo file writ  
| | > kcmgtsf(0, 0, ...) kernel cache misc. get time  
| | > kcrfw_slave_adaptive_updatemode(0, 0x600161a0, ...) kernel cache redo file writ  
| | > kcrfw_defer_write(0, 0x600161a0, ...) kernel cache redo file writ  
| | > skgstmGetEpochTs(0, 0x600161a0, ...) o/s dependent kernel gener  
| | > kcsnew3(0x600113b8, 0x7ffc3dd10d28, ...) kernel cache scn management  


---

| | > ktfwtsm(0x397f5075, 0x7ffc3dd10d28, ...) kernel transaction flashbac  
| | > kcn_stm_write(0x7ffc3dd10d28, 0x782523, ...) kernel cache nologging(?)  
| | > kcrfw_redo_write_initpic(0x77eced90, 0x7ffc3dd10d28, ...) kernel cache redo file w  
| | > kscn_to_ub8_impl(0x7ffc3dd10d28, 0, ...) kernel service convert SCN  
| | > kcrfw_gather_lwn(0x7ffc3dd10d68, 0x77eced90, ...) kernel cache redo file writ  
| | > krsh_trace(0x1000, 0x200, ...) kernel recover standby/data  
| | > kspgip(0x71e, 0x1, ...) kernel service parameter ?  
| | > kcrfw_slave_queue_flush_internal(0x1, 0x7ae0c7c0, ...) kernel cache redo file writ  
| | > kcrfw_do_null_write(0, 0, ...) kernel cache redo file writ
```

KSBCTI - 2

```
| | > kcsnew3(0x600113b8, 0x7ffc3dd10d28, ...) kernel cache scn man. get new scn  
| | | > kcsnew8(0x600113b8, 0x7ffc3dd10b70, ...) kernel cache scn man. get new scn  
| | | | > kslgetl(0x60049800, 0x1, ...) kernel service lock man. get latch  
| | | | < kslgetl+0x000000000012f returns: 0x1 ?  
| | | | > kslfre(0x60049800, 0x1, ...) kernel service lock man. free latch  
| | | | < kslfre+0x00000000001e2 returns: 0  
| | | < kcsnew8+0x0000000000117 returns: 0  
| | | > ub8_to_kscn_imp(0x7824dd, 0x7ffc3dd10d28, ...) convert ub8 to kernel SCN  
| | | < ub8_to_kscn_imp+0x0000000000031 returns: 0  
| | < kcsnew3+0x000000000006f returns: 0x8000
```

```
SQL> @flba 60049800
```

ADDR	NAME
0000000060049800	lgwr LWN SCN

KSBCTI - 2

```
kcsnew8+107:0x60016274 (fixed sga|var:kcrfsg_+76 (indx:0)X$KCRFWS+68):W:4:0x1/1()
kcsnew8+118:0x60016270 (fixed sga|var:kcrfsg_+72 (indx:0)X$KCRFWS+64):R:4:0x145f/5215()
kcsnew8+118:0x60016270 (fixed sga|var:kcrfsg_+72 (indx:0)X$KCRFWS+64):W:4:0x1460/5216()
kcsnew8+125:0x600113b8 (fixed sga|var:kcs_gscn_+0):R:8:0xf9649/1021513()
kcsnew8+131:0x00007ffd5bb83fe0():W:4:0/0()
kcsnew8+140:0x00007ffd5bb84012():W:1:0/0()
kcsnew8+149:0x600113b8 (fixed sga|var:kcs_gscn_+0):R:8:0xf9649/1021513()
kcsnew8+149:0x600113b8 (fixed sga|var:kcs_gscn_+0):W:8:0xf964a/1021514()
kcsnew8+165:0x00007ffd5bb83fe0():R:4:0/0()
kcsnew8+184:0x00007f4bd6d27e08():R:8:0x600113b8/1610683320 (fixed sga|var:kcs_gscn_+0)
kcsnew8+193:0x00007f4bd6d26e90():R:8:0x6dfd1508/1845302536 (shared pool|SQLA^ad2fc000)
kcsnew8+212:0x6dfd1d28 (shared pool|SQLA^ad2fc000):R:8:0x667bbd28/1719385384 (shared pool
408.kgghte,duration 1,cls freeabl+1704 shared pool|KGLH0^24d767d4,duration 1,cls freeabl
kcsnew8+219:0x667bc5e0 (shared pool|KGLH0^24d767d4 DS:408.kgghte,duration 1,cls freeabl-
KGLH0^24d767d4,duration 1,cls freeabl+3976 ):R:8:0x1461/5217()
kcsnew8+219:0x667bc5e0 (shared pool|KGLH0^24d767d4 DS:408.kgghte,duration 1,cls freeabl-
KGLH0^24d767d4,duration 1,cls freeabl+3976 ):W:8:0x1462/5218()
kcsnew8+223:0x00007ffd5bb840f0():W:8:0xf964a/1021514()
kcsnew8+232:0x60016260 (fixed sga|var:kcrfsg_+56 x_kcrfws.lwn_scn+0):W:8:0xf964a/1021514
kcsnew8+238:0x60016274 (fixed sga|var:kcrfsg_+76):W:4:0/0()
kcsnew8+246:0x60016288 (fixed sga|var:kcrfsg_+96 *lgwr lwn scn latch+0):R:8:0x60049800/1
Size(pgsz:2048k)|+301056 fixed sga|(parent)latch:lgwr LWN SCN+0 fixed sga|var:lwn_scn_
kcsnew8+250:0x00007ffd5bb83fc8():W:8:0x10e3ed3f/283372863()
```

KSBCTI - 2

```
kcsnew8+107:0x60016274 (fixed sga|var:kcrfsg_+76 (indx:0)X$KCRFWS+68):W:4:0x1/1 ()
kcsnew8+118:0x60016270 (fixed sga|var:kcrfsg_+72 (indx:0)X$KCRFWS+64):R:4:0x145f/5215 ()
kcsnew8+118:0x60016270 (fixed sga|var:kcrfsg_+72 (indx:0)X$KCRFWS+64):W:4:0x1460/5216 ()
kcsnew8+125:0x600113b8 (fixed sga|var:kcs_gscn_+0):R:8:0xf9649/1021513 ()
kcsnew8+131:0x00007ffd5bb83fe0 ():W:4:0/0 ()
kcsnew8+140:0x00007ffd5bb84012 ():W:1:0/0 ()
kcsnew8+149:0x600113b8 (fixed sga|var:kcs_gscn_+0):R:8:0xf9649/1021513 ()
kcsnew8+149:0x600113b8 (fixed sga|var:kcs_gscn_+0):W:8:0xf964a/1021514 ()
kcsnew8+165:0x00007ffd5bb83fe0 ():R:4:0/0 ()
kcsnew8+184:0x00007f4bd6d27e08 ():R:8:0x600113b8/1610683320 (fixed sga|var:kcs_gscn_+0)
kcsnew8+193:0x00007f4bd6d26e90 ():R:8:0x6dfd1508/1845302536 (shared pool|SQLA^ad2fc000)
kcsnew8+212:0x6dfd1d28 (shared pool|SQLA^ad2fc000):R:8:0x667bbd28/1719385384 (shared pool
408.kgghte,duration 1,cls freeabl+1704 shared pool|KGLH0^24d767d4,duration 1,cls freeabl
kcsnew8+219:0x667bc5e0 (shared pool|KGLH0^24d767d4 DS:408.kgghte,duration 1,cls freeabl-
KGLH0^24d767d4,duration 1,cls freeabl+3976 ):R:8:0x1461/5217 ()
kcsnew8+219:0x667bc5e0 (shared pool|KGLH0^24d767d4 DS:408.kgghte,duration 1,cls freeabl-
KGLH0^24d767d4,duration 1,cls freeabl+3976 ):W:8:0x1462/5218 ()
kcsnew8+223:0x00007ffd5bb840f0 ():W:8:0xf964a/1021514 ()
kcsnew8+232:0x60016260 (fixed sga|var:kcrfsg_+56 x_kcrfws.lwn_scn+0):W:8:0xf964a/1021514
kcsnew8+238:0x60016274 (fixed sga|var:kcrfsg_+76):W:4:0/0 ()
kcsnew8+246:0x60016288 (fixed sga|var:kcrfsg_+96 *lgwr lwn scn latch+0):R:8:0x60049800/1
Size(pgsz:2048k)|+301056 fixed sga|(parent)latch:lgwr LWN SCN+0 fixed sga|var:lwn_scn_
kcsnew8+250:0x00007ffd5bb83fc8 ():W:8:0x10e3ed3f/283372863 ()
```

KSBCTI - 2

```
awk '/^\ \|\ \x3e\ kcrfw_redo_write_driver/, /^\ \|\ \x3c\ kcrfw_redo_write_driver/'  
lgwr_full_cycle_stripped.txt | awk '/^\ \|\ \x3e\ / || /^\ \|\ \|\ \x3e/'
```

```
| > kcrfw_redo_write_driver(0, 0, ...) kernel cache redo file writ  
| | > kcrfw_handle_member_write_errors(0, 0, ...) kernel cache redo file writ  
| | > kcmgtsf(0, 0, ...) kernel cache misc. get time  
| | > kcrfw_slave_adaptive_updatemode(0, 0x600161a0, ...) kernel cache redo file writ  
| | > kcrfw_defer_write(0, 0x600161a0, ...) kernel cache redo file writ  
| | > skgstmGetEpochTs(0, 0x600161a0, ...) o/s dependent kernel gener  
| | > kcsnew3(0x600113b8, 0x7ffc3dd10d28, ...) kernel cache scn management  


---

| | > ktfwtsm(0x397f5075, 0x7ffc3dd10d28, ...) kernel transaction flashbac  
| | > kcn_stm_write(0x7ffc3dd10d28, 0x782523, ...) kernel cache nologging(?)  
| | > kcrfw_redo_write_initpic(0x77eced90, 0x7ffc3dd10d28, ...) kernel cache redo file w  
| | > kscn_to_ub8_impl(0x7ffc3dd10d28, 0, ...) kernel service convert SCN  
| | > kcrfw_gather_lwn(0x7ffc3dd10d68, 0x77eced90, ...) kernel cache redo file writ  
| | > krsh_trace(0x1000, 0x200, ...) kernel recover standby/data  
| | > kspgip(0x71e, 0x1, ...) kernel service parameter ?  
| | > kcrfw_slave_queue_flush_internal(0x1, 0x7ae0c7c0, ...) kernel cache redo file writ  
| | > kcrfw_do_null_write(0, 0, ...) kernel cache redo file writ
```

KSBCTI - 2

```
| | > kcrfw_gather_lwn(0x7ffc3dd10d68, 0x77eced90, ...) kernel cache redo file write gat
| | | > kslgetl(0x76fe0c10, 0x1, ...) kernel service lock man. get lat
| | | < kslgetl+0x000000000012f returns: 0x1
| | | > kcrfw_gather_strand(0x7ffc3dd10d68, 0, ...) kernel cache redo file write gat
| | | < kcrfw_gather_strand+0x0000000000c2 returns: 0
| | | > kslfre(0x76fe0c10, 0x118b2, ...) kernel service lock man. free la
| | | < kslfre+0x00000000001e2 returns: 0
| | | > kslgetl(0x76fe0cb0, 0x1, ...)
| | | < kslgetl+0x000000000012f returns: 0x1
| | | > kcrfw_gather_strand(0x7ffc3dd10d68, 0x1, ...)
| | | < kcrfw_gather_strand+0x0000000000c2 returns: 0
| | | > kslfre(0x76fe0cb0, 0x19, ...)
| | | < kslfre+0x00000000001e2 returns: 0
| | < kcrfw_gather_lwn+0x000000000065c returns: 0xffffffff
```

KSBCTI - 2

```
kcrfw_gather_lwn+76:0x600169b8(fixed sga|var:kcrf max strands +0):R:4:0x2/2()  
kcrfw_gather_lwn+223:0x60016570(fixed sga|var:kcrfsg_+840):R:8:0x76fe0c10/1996360720(sh  
(child)latch#1:redo allocation+0 shared pool|permanent memor,duration 1,cls perm+161208  
kcrfw_gather_lwn+230:0x60016228(fixed sga|var:kcrfsg_+0|pointer:shared pool redo struct  
:R:8:0x76fdf388/1996354440(shared pool|KCRFA+0 shared pool|permanent memor,duration 1,c  
redo allocation latch child #1 is gotten  
kcrfw_gather_strand+36:0x76fdf458(shared pool|KCRFA+208 LAST_BUF_WRITTEN+0  
shared pool|permanent memor,duration 1,cls perm+16114776 ):R:4:0x18f18/102168()  
kcrfw_gather_strand+86:0x76fdf3a8(shared pool|KCRFA+32 NEXT_BUF_NUM_KCRFA_CLN+0  
shared pool|permanent memor,duration 1,cls perm+16114600 ):R:4:0x18f18/102168()  
kcrfw_gather_strand+121:0x76fdf45c(shared pool|KCRFA+212 LAST_BUF_GATHERED_KCRFA+0  
shared pool|permanent memor,duration 1,cls perm+16114780 ):R:4:0x18f18/102168()  
redo allocation latch child #1 is freed  
kcrfw_gather_lwn+1436:0x77ecee70(shared pool|permanent memor,duration 1,cls perm+245172  
:W:8:0/0()  
kcrfw_gather_lwn+1443:0x77eceeac(shared pool|permanent memor,duration 1,cls perm+245178  
:W:4:0x1/1()  
kcrfw_gather_lwn+1451:0x77ecee70(shared pool|permanent memor,duration 1,cls perm+245178  
:W:4:0/0()  
kcrfw_gather_lwn+1455:0x77ecee78(shared pool|permanent memor,duration 1,cls perm+245179  
:W:4:0/0()
```

fixed sga variable kcrfsg_ for version 12.2.0.1.170814

offset	address	length	annotation
0	0x60016228	8	*KCRFA struct
8	0x60016230	4	x\$kcrfws.next_blk
12	0x60016234	4	x\$kcrfws.last_blk
20	0x6001623C	8	x\$kcrfws.last_write_scn_time
56	0x60016260	8	x\$kcrfws.lwn_scn
72	0x60016270	4	lwn number
76	0x60016274	4	lwn modify flag
96	0x60016288	8	*lgwr lwn scn latch
104	0x60016290	8	x\$kcrfws.on_disk_scn
152	0x600162C0	4	x\$kcrfws.last_write_scn
160	0x600162C8	4	log sequence number
164	0x600162CC	4	rba current log block
176	0x600162D8	4	rba modify flag
184	0x600162E0	8	*consistent rba latch
288	0x60016348	8	x\$kcrfws.real_redo_scn
340	0x6001637C	8	x\$kcrfws.real_write_time
392	0x600163B0	8	log block size
464	0x600163F8	8	number redo copy latches
472	0x60016400	8	*redo copy latch
488	0x60016410	8	*redo copy usage status memory area
516	0x6001642C	8	messages; is log writer posted 1/yes 0/no
840	0x60016570	8	*redo allocation latch

shared pool struct kcrfa for version 12.2.0.1.170814

offset	address	length	annotation
24	0x76fdf3a0	8	pnext_buf_kcrfa_cln
32	0x76fdf3a8	4	next_buf_num_kcrfa_cln
40	0x76fdf3b0	8	bytes_in_buf_kcrfa_cln
208	0x76fdf458	4	last_buf_written
212	0x76fdf45c	4	last_buf_gathered_kcrfa
248	0x76fdf480	4	total_bufs_kcrfa

KSBCTI - 2

```
| | > kcrfw_gather_lwn(0x7ffc3dd10d68, 0x77eced90, ...) kernel cache redo file write gat
| | | > kslgetl(0x76fe0c10, 0x1, ...) kernel service lock man. get lat
| | | < kslgetl+0x000000000012f returns: 0x1
| | | > kcrfw_gather_strand(0x7ffc3dd10d68, 0, ...) kernel cache redo file write gat
| | | < kcrfw_gather_strand+0x0000000000c2 returns: 0
| | | > kslfre(0x76fe0c10, 0x118b2, ...) kernel service lock man. free la
| | | < kslfre+0x00000000001e2 returns: 0
| | | > kslgetl(0x76fe0cb0, 0x1, ...)
| | | < kslgetl+0x000000000012f returns: 0x1
| | | > kcrfw_gather_strand(0x7ffc3dd10d68, 0x1, ...)
| | | < kcrfw_gather_strand+0x0000000000c2 returns: 0
| | | > kslfre(0x76fe0cb0, 0x19, ...)
| | | < kslfre+0x00000000001e2 returns: 0
| | < kcrfw_gather_lwn+0x000000000065c returns: 0xffffffff
```

KSBCTI - 2

```
awk '/^\ \|\ \x3e\ kcrfw_redo_write_driver/, /^\ \|\ \x3c\ kcrfw_redo_write_driver/'  
lgwr_full_cycle_stripped.txt | awk '/^\ \|\ \x3e\ / || /^\ \|\ \|\ \x3e/'
```

```
| > kcrfw_redo_write_driver(0, 0, ...) kernel cache redo file writ  
| | > kcrfw_handle_member_write_errors(0, 0, ...) kernel cache redo file writ  
| | > kcmgtsf(0, 0, ...) kernel cache misc. get time  
| | > kcrfw_slave_adaptive_updatemode(0, 0x600161a0, ...) kernel cache redo file writ  
| | > kcrfw_defer_write(0, 0x600161a0, ...) kernel cache redo file writ  
| | > skgstmGetEpochTs(0, 0x600161a0, ...) o/s dependent kernel gener  
| | > kcsnew3(0x600113b8, 0x7ffc3dd10d28, ...) kernel cache scn management  
| | > ktfwtsm(0x397f5075, 0x7ffc3dd10d28, ...) kernel transaction flashbac  
| | > kcn_stm_write(0x7ffc3dd10d28, 0x782523, ...) kernel cache ??  
| | > kcrfw_redo_write_initpic(0x77eced90, 0x7ffc3dd10d28, ...) kernel cache redo file w  
| | > kscn_to_ub8_impl(0x7ffc3dd10d28, 0, ...) kernel service convert SCN  
| | > kcrfw_gather_lwn(0x7ffc3dd10d68, 0x77eced90, ...) kernel cache redo file writ  


---

| | > krsh_trace(0x1000, 0x200, ...) kernel recover standby/data  
| | > kspgip(0x71e, 0x1, ...) kernel service parameter ?  
| | > kcrfw_slave_queue_flush_internal(0x1, 0x7ae0c7c0, ...) kernel cache redo file writ  
| | > kcrfw_do_null_write(0, 0, ...) kernel cache redo file writ
```

KSBCTI - 2

```
awk '/^\ \|\ \|\ \x3e\ kcrfw_do_null_write/,/^\ \|\ \|\ \x3c\ kcrfw_do_null_write/'  
lgwr_full_cycle_stripped.txt |  
awk '/^\ \|\ \|\ \x3e/||/^\ \|\ \|\ \|\ \x3e/'
```

```
| | > kcrfw_do_null_write(0, 0, ...)      kernel cache redo file write only increase  
| | | > kcrfw_slave_phase_batchdo(0, 0, ...) kernel cache redo file write ??  
| | | > kcrfw_post(0, 0, ...)           kernel cache redo file write post processing
```

KSBCTI - 2

```
| | | > kcrfw_post(0, 0, ...) kernel cache redo file
| | | | > kcrfw_slave_single_getactivegroup(0, 0, ...) kernel cache redo file
| | | | > kspGetInstType(0x1, 0x1, ...) kernel service paramet
| | | | > kcrfw_slave_phase_enter(0x1, 0x81, ...) kernel cache redo file
| | | | > kcscu8(0x60016290, 0x7ffc3dd10a98, ...) kernel service scn man
| | | | > kcsaj8(0x60016290, 0x7ffc3dd10a38, ...) kernel service scn man
| | | | > kcrfw_slave_phase_exit(0x1, 0x81, ...) kernel cache redo file
| | | | > kslpsemf(0x97, 0, ...) kernel service lock ma
| | | | > kcrfw_slave_barrier_nonmasterwait(0x769a5628, 0x4, ...) kernel cache redo file
```

LOG WRITER - IDLE

Conclusion on the idle loop of the log writer:

- The log writer process calculates to have a 3 second cycle of processing after it executed its tasks.
- Then sleeps on a semaphore (semtimedop).
- The work in the idle cycle takes very little CPU, although it needs to get running after being woken.
- Some latches are taken: messages, lgwr LWN SCN, redo allocation. All in willing to wait mode.
- Some latches are taken shared: KTF sga latch, Nologging Standby SCN Time Map cache.

- The only latch that could be considered potentially busy is the redo allocation latch.

LOG WRITER - DML

When DML (insert, update, insert, delete, upsert) in regular mode is executed, a **FOREGROUND PROCESS** executes:

- gets a block in current mode (kcbgcur)
- it prepares the change vectors (ktugur, kcopcv)
- obtains a random redo copy latch in immediate mode (kslgetl)
- obtains a redo allocation latch in immediate mode of a public redo strand (kslgetl)
- modifies KCRFA struct for the strand used (kcrfw_redo_gen_ext)
- frees the redo allocation latch (kslfre)
- copies the change vectors into a public redo strand (kcrfw_copy_cv)
- applies changes to buffers (in kcoapl: block type and action specific functions)
- frees the redo copy latch (kslfre)

The important bit is the public redo strands are filled with change vectors. **Mind the absence of commit!**

LOG WRITER - DML

The absence of commit means the log writer is not signalled (to write).

This means:

- the log writer process just times out on the semaphore.
- performs the just described work cycle.
- of course, it will encounter the change vectors copied in the public strand by looking at KCRFA.
 - so the first time the log writer understands it must write something is in `kcrfw_gather_lwn`, in the function `kcrfw_gather_strand`

KSBCTI - 2

```
| | > kcrfw_gather_lwn(0x7fff7f344de8, 0x699b4428, ...)  
| | | > kslgetl(0x6abe4148, 0x1, ...)  
| | | < kslgetl+0x00000000012f returns: 0x1  
| | | > kcrfw_gather_strand(0x7fff7f344de8, 0, ...)  
| | | < kcrfw_gather_strand+0x0000000000c2 returns: 0x1  
| | | > kslfre(0x6abe4148, 0x20001, ...)  
| | | < kslfre+0x0000000001e2 returns: 0  
| | | > kslgetl(0x6abe41e8, 0x1, ...)  
| | | < kslgetl+0x00000000012f returns: 0x1  
| | | > kcrfw_gather_strand(0x7fff7f344de8, 0x1, ...)  
| | | < kcrfw_gather_strand+0x0000000000c2 returns: 0  
| | | > kslfre(0x6abe41e8, 0x1ffff, ...)  
| | | < kslfre+0x0000000001e2 returns: 0  
| | < kcrfw_gather_lwn+0x000000000065c returns: 0
```

KSBCTI - 2

```
| > kcrfw_redo_write_driver(0, 0, ...)  
| | > kcrfw_handle_member_write_errors(0, 0, ...)  
| | > kcmgtsf(0, 0, ...)  
| | > kcrfw_slave_adaptive_updatemode(0, 0x600161a0, ...)  
| | > kcrfw_defer_write(0, 0x600161a0, ...)  
| | > skgstmGetEpochTs(0, 0x600161a0, ...)  
| | > kcsnew3(0x600113b8, 0x7fff7f344da8, ...)  
| | > ktfwtsm(0x3a27a6a7, 0x7fff7f344da8, ...)  
| | > kcn_stm_write(0x7fff7f344da8, 0x7ceb23, ...)  
| | > kcrfw_redo_write_initpic(0x699b4428, 0x7fff7f344da8, ...)  
| | > kscn_to_ub8_impl(0x7fff7f344da8, 0, ...)  
| | > kcrfw_gather_lwn(0x7fff7f344de8, 0x699b4428, ...)  
| | > kcrfw_slave_adaptive_updatestats(0x6abe41e8, 0x200, ...)  
| | > kcrfw_redo_write(0, 0, ...)
```

KSBCTI - 2

```
| | > kcrfw_redo_write(0, 0, ...) kernel cache redo file write red
| | | > kcrfw_slave_single_getactivegroup(0, 0, ...) kernel cache redo file write ??
| | | > kcrfw_slave_single_getstrands(0x7fff7f344be8, 0x7fff7f344bec, ...) kernel cache
| | | > kcrfw_wait_for_fg_copy(0x699b4428, 0, ...) kernel cache redo file write rea
| | | > kcrfw_have_non_idle_strand(0x699b4428, 0, ...) kernel cache redo file write gat
| | | > kcrfw_update_and_bcast_real_redo_scn(0x699b4428, 0x976a8000, ...) kernel cache r
| | | > kcrfw_update_redo_blk_hdrs(0x699b4428, 0, ...) kernel cache redo file write rea
| | | > kcrfw_update_strand_hdrs(0x7fff7f344da8, 0x699b4428, ...) kernel cache redo file
| | | > kcrfw_cal_target_rba(0x5, 0x976a8050, ...) kernel cache redo file write cal
| | | > skgstmGetEpochTs(0x6004b780, 0x1, ...) o/s dependent kernel generic ser
| | | > kcrfw_do_write(0x699b4428, 0x1, ...) kernel cache redo file write exe
| | | > kcrfw_slave_barrier_masterwait(0x6a9a0670, 0x3, ...) kernel cache redo file writ
| | | > kcrfw_postprocess_write(0x699b4428, 0x3, ...) kernel cache redo file write ??
| | | > kcrfw_post(0, 0, ...) kernel cache redo file write pos
| | | > kcrfw_stats_prof_print(0x699b4428, 0, ...) kernel cache redo file write ??
```

KCRFW_WAIT_FOR_FG_COPY

kcrfw_wait_for_fg_copy+66:0x600163f8(fixed sga|var:kcrfsg_+464):R:4:0x8/8()

...

kcrfw_wait_for_fg_copy+132:0x60016400(fixed sga|var:kcrfsg_+472 pointer:redo copy latch
:R:8:0x76fdea30/1996352048(shared pool|(child)latch#1:redo copy+0 shared pool|permanent

kcrfw_wait_for_fg_copy+142:0x60016410(fixed sga|var:kcrfsg_+488)

:R:8:0x76fdf0f0/1996353776(shared pool|permanent memor,duration 1,cls perm+16113904)

kcrfw_wait_for_fg_copy+149:0x00007fff2b352568():W:1:0/0()

kcrfw_wait_for_fg_copy+153:0x00007fff2b352560()

:W:8:0x76fdea30/1996352048(shared pool|(child)latch#1:redo copy+0 shared pool|permanent

kcrfw_wait_for_fg_copy+157:0x00007fff2b352558():W:4:0/0()

kcrfw_wait_for_fg_copy+161:0x76fdf0f0(shared pool|permanent memor,duration 1,cls perm+1

:R:4:0xffffffff/4294967295(??)

kcrfw_wait_for_fg_copy+169:0x76fdf0f4(shared pool|permanent memor,duration 1,cls perm+1

:R:4:0xffffffff/4294967295(??)

kcrfw_wait_for_fg_copy+195:0x600163f8(fixed sga|var:kcrfsg_+464):R:4:0x8/8()

kcrfw_wait_for_fg_copy+132:0x60016400(fixed sga|var:kcrfsg_+472 pointer:redo copy latch
:R:8:0x76fdea30/1996352048(shared pool|(child)latch#1:redo copy+0 shared pool|permanent

kcrfw_wait_for_fg_copy+142:0x60016410(fixed sga|var:kcrfsg_+488)

:R:8:0x76fdf0f0/1996353776(shared pool|permanent memor,duration 1,cls perm+16113904⁴⁴)

kcrfw_wait_for_fg_copy+149:0x00007fff2b352568():W:1:0/0()

```
...
kcrfw_wait_for_fg_copy+132:0x60016400 (fixed sga|var:kcrfsg_+472 pointer:redo copy latch
:R:8:0x76fdea30/1996352048 (shared pool|(child)latch#1:redo copy+0 shared pool|permanent
kcrfw_wait_for_fg_copy+142:0x60016410 (fixed sga|var:kcrfsg_+488)
:R:8:0x76fdf0f0/1996353776 (shared pool|permanent memor,duration 1,cls perm+16113904 )
kcrfw_wait_for_fg_copy+149:0x00007fff2b352568():W:1:0/0()
kcrfw_wait_for_fg_copy+153:0x00007fff2b352560()
:W:8:0x76fdea30/1996352048 (shared pool|(child)latch#1:redo copy+0 shared pool|permanent
kcrfw_wait_for_fg_copy+157:0x00007fff2b352558():W:4:0/0()
kcrfw_wait_for_fg_copy+161:0x76fdf0f0 (shared pool|permanent memor,duration 1,cls perm+1
:R:4:0xffffffff/4294967295(??)
kcrfw_wait_for_fg_copy+169:0x76fdf0f4 (shared pool|permanent memor,duration 1,cls perm+1
:R:4:0xffffffff/4294967295(??)
kcrfw_wait_for_fg_copy+195:0x600163f8 (fixed sga|var:kcrfsg_+464):R:4:0x8/8()
kcrfw_wait_for_fg_copy+132:0x60016400 (fixed sga|var:kcrfsg_+472 pointer:redo copy latch
:R:8:0x76fdea30/1996352048 (shared pool|(child)latch#1:redo copy+0 shared pool|permanent
kcrfw_wait_for_fg_copy+142:0x60016410 (fixed sga|var:kcrfsg_+488)
:R:8:0x76fdf0f0/1996353776 (shared pool|permanent memor,duration 1,cls perm+16113904 )
kcrfw_wait_for_fg_copy+149:0x00007fff2b352568():W:1:0/0()
kcrfw_wait_for_fg_copy+153:0x00007fff2b352560():W:8:0x76fdeb08/1996352264 (
shared pool|(child)latch#2:redo copy+0 shared pool|permanent memor,duration 1,cls perm-
kcrfw_wait_for_fg_copy+157:0x00007fff2b352558():W:4:0x1/1()
kcrfw_wait_for_fg_copy+161:0x76fdf0f8 (shared pool|permanent memor,duration 1,cls perm+1
:R:4:0xffffffff/4294967295(??)
kcrfw_wait_for_fg_copy+169:0x76fdf0fc (shared pool|permanent memor,duration 1,cls perm+1
:R:4:0xffffffff/4294967295(??)
```

KSBCTI - 2

```
| | > kcrfw_redo_write(0, 0, ...) kernel cache redo file write rec
| | | > kcrfw_slave_single_getactivegroup(0, 0, ...) kernel cache redo file write ??
| | | > kcrfw_slave_single_getstrands(0x7fff7f344be8, 0x7fff7f344bec, ...) kernel cache
| | | > kcrfw_wait_for_fg_copy(0x699b4428, 0, ...) kernel cache redo file write rea
| | | > kcrfw_have_non_idle_strand(0x699b4428, 0, ...) kernel cache redo file write gat
| | | > kcrfw_update_and_bcast_real_redo_scn(0x699b4428, 0x976a8000, ...) kernel cache r
| | | > kcrfw_update_redo_blk_hdrs(0x699b4428, 0, ...) kernel cache redo file write rea
| | | > kcrfw_update_strand_hdrs(0x7fff7f344da8, 0x699b4428, ...) kernel cache redo file
| | | > kcrfw_cal_target_rba(0x5, 0x976a8050, ...) kernel cache redo file write cal
| | | > skgstmGetEpochTs(0x6004b780, 0x1, ...) o/s dependent kernel generic ser
| | | > kcrfw_do_write(0x699b4428, 0x1, ...) kernel cache redo file write exe
| | | > kcrfw_slave_barrier_masterwait(0x6a9a0670, 0x3, ...) kernel cache redo file writ
| | | > kcrfw_postprocess_write(0x699b4428, 0x3, ...) kernel cache redo file write ??
| | | > kcrfw_post(0, 0, ...) kernel cache redo file write pos
| | | > kcrfw_stats_prof_print(0x699b4428, 0, ...) kernel cache redo file write ??
```

KSBCTI - 2

```
| | > kcrfw_do_write(0x77eced90, 0x1, ...) kernel cache redo file write e
| | | > kcrfw_slave_single_getactivegroup(0x77eced90, 0x1, ...) kernel cache redo file w
| | | > kcrfw_slave_single_getstrands(0x7ffc3dd118e4, 0x7ffc3dd118e8, ...) kernel cache
| | | > ksfdgrsz(0x1, 0, ...) kernel service functions disk
| | | > kcrfw_slave_single_strand2slvid(0, 0, ...) kernel cache redo file write 1
| | | > kcrfw_slave_single_strand2slvid(0x1, 0, ...) kernel cache redo file write 1
| | | > kghstack_alloc(0x7f2c23b9e9a0, 0x8, ...) kernel generic heap manager al
| | | > ksfd_aio_setobjinfo(0x71717da0, 0x76b423e8, ...) kernel service functions disk
| | | > ksfdrqfill1(0x71717da0, 0x76b423e8, ...) kernel service functions disk
| | | > skgstmGetEpochTs(0x76b423e8, 0x71717f00, ...) o/s dependent kernel generic s
| | | > rfafoAssessLag(0x7ffc3dd116a0, 0x2050c6a4e6528, ...) data guard(?)
| | | > ksfdbio(0x7f2c1e502000, 0x1, ...) kernel service functions disk
| | | > skgstmGetEpochTs(0x7f2c23b9e9a0, 0x7f2c1e502028, ...) o/s dependent kernel gener
| | | > ksfdckerr(0x71717da0, 0x2050c6a4f63be, ...) kernel service functions disk
| | | > kghstack_free(0x7f2c23b9e9a0, 0x7f2c1e502000, ...) kernel generic heap manager fr
| | | > krsht_trace(0x1000, 0x7f2c1e502000, ...) kernel recovery standby/datagu
```

KSBCTI - 2

```
| | | > ksfdbio(0x7f2c1e502000, 0x1, ...) kernel service functions disk I/O
| | | | > kskiorm(0x7f2c1e502000, 0x1, ...) kernel service resource manager
| | | | > kspgip(0x17d, 0x1, ...) kernel service parameter ??
| | | | > kspgip(0x17c, 0x1, ...) kernel service parameter ??
| | | | > ksfdss_fob_is_snap(0x76b423e8, 0x7ae0c7c0, ...) kernel service functions disk I/O
| | | | > kghstack_alloc(0x7f2c23b9e9a0, 0x8, ...) kernel generic heap manager allocation
| | | | > ksfd_update_iostatsbytes(0, 0x76b76930, ...) kernel service functions disk I/O
| | | | > kslwtbctx(0x7ffc3dd11700, 0x42, ...) kernel service lock man. wait resource
| | | | > ksfd_sbinfo(0x7f2c1e502028, 0x1, ...) kernel service functions disk I/O
| | | | > kslwtectx(0x7ffc3dd11700, 0x200, ...) kernel service lock man. wait resource
| | | | > kgs_lwait_resource_time_waited_usecs(0x7f2c23b9e9a0, 0x7ffc3dd11700, ...) kernel service lock man. wait resource
| | | | > ksfdbio_updatestats(0x500, 0, ...) kernel service functions disk I/O
| | | | > kghstack_free(0x7f2c23b9e9a0, 0x7f2c1e502028, ...) kernel generic heap manager
```


KSBCTI - 2

```
| | | | > ksfd_sbio(0x7f2c1e502028, 0x1, ...) kernel service functions disk IO
| | | | | > ksfd_osdrqfil(0x71717da0, 0x76b423e8, ...) kernel service functions disk IO
| | | | | > ksfd_addtoissueq(0x71717da0, 0x200, ...) kernel service functions disk IO
| | | | | > ksfdgo(0x500, 0x81e, ...) kernel service functions disk IO
| | | | | > ksfdwtio(0x1, 0x500, ...) kernel service functions disk IO
| | | | | > ksfd_chkio(0x71717da0, 0x9, ...) kernel service functions disk IO
```

KSBCTI - 2

```
| | | | | > ksfdgo(0x500, 0x81e, ...) kernel service functions
| | | | | | > ksfd skgfqio(0x7f2c23b04080, 0x9, ...) kernel service functions
| | | | | | | > skgfqio(0x7f2c23ba8860, 0x7f2c23b04080, ...) o/s dependent kernel gen
| | | | | | | | > skgfrvldtrq(0x7f2c23b04080, 0x9, ...) o/s dependent kernel gen
| | | | | | | | < skgfrvldtrq+0x0000000000297 returns: 0
| | | | | | | | > sltrgftime64(0xda316600, 0x71717f88, ...) o/s dependent call time
| | | | | | | | | > clock_gettime@plt(0x1, 0x7ffc3dd0c420, ...)
| | | | | | | | | | > clock_gettime(0x1, 0x7ffc3dd0c420, ...)
| | | | | | | | | | < clock_gettime+0x00000000000069 returns: 0
| | | | | | | | | | < clock_gettime+0x0000000000003a returns: 0
| | | | | | | | | | < sltrgftime64+0x0000000000004c returns: 0x9cdee0a03
| | | | | | | | | > skgfr_lio_listio64(0x7f2c23ba8860, 0x1, ...) o/s dependent kernel gen
| | | | | | | | | | > io_submit@plt(0x7f2c23bef000, 0x1, ...)
| | | | | | | | | | < io_submit+0x00000000000007 returns: 0x1
| | | | | | | | | | < skgfr_lio_listio64+0x000000000000131 returns: 0
| | | | | | | | | < skgfqio+0x000000000000369 returns: 0
| | | | | | | | < ksfd_skgfio+0x0000000000001f9 returns: 0
| | | | | | | < ksfdgo+0x000000000000135 returns: 0
```

KSBCTI - 2

```
| | | | > ksfd_sbio(0x7f2c1e502028, 0x1, ...) kernel service functions disk IO
| | | | | > ksfd_osdrqfil(0x71717da0, 0x76b423e8, ...) kernel service functions disk IO
| | | | | > ksfd_addtoissueq(0x71717da0, 0x200, ...) kernel service functions disk IO
| | | | | > ksfdgo(0x500, 0x81e, ...) kernel service functions disk IO
| | | | | > ksfdwtio(0x1, 0x500, ...) kernel service functions disk IO
| | | | | > ksfd_chkio(0x71717da0, 0x9, ...) kernel service functions disk IO
```

KSBCTI - 2

```
| | | | | > ksfdwtio(0x1, 0x500, ...)

---

| | | | | > skgfrwat(0x7ffc3dd11480, 0x7f2c23ba8860, ...)  
| | | | | > skgfospo(0x7ffc3dd11480, 0x7f2c23ba8860, ...)  
| | | | | <> skgfrliopo(0x7ffc3dd11480, 0x7f2c23ba8860, ...)  
...  
| | | | | | > io_getevents@plt(0x7f2c23bef000, 0x1, ...)  
| | | | | | <> io_queue_run+0x00000000000050(0x7f2c23bef000, 0x1, ...)  
| | | | | | < io_queue_run+0x00000000000065 returns: 0x1  
| | | | | | > sltrgftime64(0x7f2c23bef000, 0x1, ...)  
| | | | | | > skgfr_error64(0x7f2c23ba8860, 0x71717fc8, ...)  
| | | | | | > skgfrciohdlr(0x7f2c23ba8860, 0x71717fa8, ...)  
| | | | | | | > skgfr_return64(0x7f2c23ba8860, 0x71717fc8, ...)  
| | | | | | > skgfgiod(0x7f2c23ba8860, 0x71717fc8, ...)  
| | | | | > kgecrs(0x7f2c23b9e9a0, 0x7f2c1e54b048, ...)  
| | | | | > ksfdpcl(0x71717da0, 0x71717f88, ...)  
| | | | | > kcbbdw(0x600161a0, 0x71717f88, ...)  
| | | | | | > ksfd_do_io_verification(0x71717da0, 0x71717f88, ...)  
| | | | | | > ksfd_gosdrq(0x71717da0, 0x71717f88, ...)  
| | | | | | > skgfnxtio(0x7ffc3dd113c0, 0x7f2c23ba8860, ...)  
| | | | | | | > skgfgiod(0x7f2c23ba8860, 0x7f2c23ba8860, ...)  
| | | | | | > ksfdgo(0x500, 0x1, ...)
```

KSBCTI - 2

```
| | | | > ksfd_sbio(0x7f2c1e502028, 0x1, ...) kernel service functions disk IO
| | | | | > ksfd_osdrqfil(0x71717da0, 0x76b423e8, ...) kernel service functions disk IO
| | | | | > ksfd_addtoissueq(0x71717da0, 0x200, ...) kernel service functions disk IO
| | | | | > ksfdgo(0x500, 0x81e, ...) kernel service functions disk IO
| | | | | > ksfdwtio(0x1, 0x500, ...) kernel service functions disk IO
| | | | | > ksfd_chkio(0x71717da0, 0x9, ...) kernel service functions disk IO
```

KSBCTI - 2

```
| | | > ksfdbio(0x7f2c1e502000, 0x1, ...) kernel service functions disk I/O
| | | | > kskiorm(0x7f2c1e502000, 0x1, ...) kernel service resource manager
| | | | > kspgip(0x17d, 0x1, ...) kernel service parameter ??
| | | | > kspgip(0x17c, 0x1, ...) kernel service parameter ??
| | | | > ksfdss_fob_is_snap(0x76b423e8, 0x7ae0c7c0, ...) kernel service functions disk I/O
| | | | > kghstack_alloc(0x7f2c23b9e9a0, 0x8, ...) kernel generic heap manager allocation
| | | | > ksfd_update_iostatsbytes(0, 0x76b76930, ...) kernel service functions disk I/O
| | | | > kslwtbctx(0x7ffc3dd11700, 0x42, ...) kernel service lock man. wait resource
| | | | > ksfd_sbinfo(0x7f2c1e502028, 0x1, ...) kernel service functions disk I/O
| | | | > kslwtctx(0x7ffc3dd11700, 0x200, ...) kernel service lock man. wait resource
| | | | > kgs_lwait_resource_time_waited_usecs(0x7f2c23b9e9a0, 0x7ffc3dd11700, ...) kernel service lock man. wait resource
| | | | > ksfdbio_updatestats(0x500, 0, ...) kernel service functions disk I/O
| | | | > kghstack_free(0x7f2c23b9e9a0, 0x7f2c1e502028, ...) kernel generic heap manager
```

KSBCTI - 2

```
| | > kcrfw_do_write(0x77eced90, 0x1, ...) kernel cache redo file write e
| | | > kcrfw_slave_single_getactivegroup(0x77eced90, 0x1, ...) kernel cache redo file w
| | | > kcrfw_slave_single_getstrands(0x7ffc3dd118e4, 0x7ffc3dd118e8, ...) kernel cache
| | | > ksfdgrsz(0x1, 0, ...) kernel service functions disk
| | | > kcrfw_slave_single_strand2slvid(0, 0, ...) kernel cache redo file write 1
| | | > kcrfw_slave_single_strand2slvid(0x1, 0, ...) kernel cache redo file write 1
| | | > kghstack_alloc(0x7f2c23b9e9a0, 0x8, ...) kernel generic heap manager al
| | | > ksfd_aio_setobjinfo(0x71717da0, 0x76b423e8, ...) kernel service functions disk
| | | > ksfdrqfill1(0x71717da0, 0x76b423e8, ...) kernel service functions disk
| | | > skgstmGetEpochTs(0x76b423e8, 0x71717f00, ...) o/s dependent kernel generic s
| | | > rfafoAssessLag(0x7ffc3dd116a0, 0x2050c6a4e6528, ...) data guard(?)
| | | > ksfdbio(0x7f2c1e502000, 0x1, ...) kernel service functions disk
| | | > skgstmGetEpochTs(0x7f2c23b9e9a0, 0x7f2c1e502028, ...) o/s dependent kernel gener
| | | > ksfdckerr(0x71717da0, 0x2050c6a4f63be, ...) kernel service functions disk
| | | > kghstack_free(0x7f2c23b9e9a0, 0x7f2c1e502000, ...) kernel generic heap manager fr
| | | > krsht_trace(0x1000, 0x7f2c1e502000, ...) kernel recovery standby/datagu
```

KSBCTI - 2

```
| | > kcrfw_redo_write(0, 0, ...) kernel cache redo file write red
| | | > kcrfw_slave_single_getactivegroup(0, 0, ...) kernel cache redo file write ??
| | | > kcrfw_slave_single_getstrands(0x7fff7f344be8, 0x7fff7f344bec, ...) kernel cache
| | | > kcrfw_wait_for_fg_copy(0x699b4428, 0, ...) kernel cache redo file write rea
| | | > kcrfw_have_non_idle_strand(0x699b4428, 0, ...) kernel cache redo file write gat
| | | > kcrfw_update_and_bcast_real_redo_scn(0x699b4428, 0x976a8000, ...) kernel cache r
| | | > kcrfw_update_redo_blk_hdrs(0x699b4428, 0, ...) kernel cache redo file write rea
| | | > kcrfw_update_strand_hdrs(0x7fff7f344da8, 0x699b4428, ...) kernel cache redo file
| | | > kcrfw_cal_target_rba(0x5, 0x976a8050, ...) kernel cache redo file write cal
| | | > skgstmGetEpochTs(0x6004b780, 0x1, ...) o/s dependent kernel generic ser
| | | > kcrfw_do_write(0x699b4428, 0x1, ...) kernel cache redo file write exe
| | | > kcrfw_slave_barrier_masterwait(0x6a9a0670, 0x3, ...) kernel cache redo file writ
| | | > kcrfw_postprocess_write(0x699b4428, 0x3, ...) kernel cache redo file write ??
| | | > kcrfw_post(0, 0, ...) kernel cache redo file write pos
| | | > kcrfw_stats_prof_print(0x699b4428, 0, ...) kernel cache redo file write ??
```



```
| | | > kcrfw_post(0x7ffc3dd122a0, 0x18, ...) kernel cache redo file write post proc
| | | | > kcrfw_slave_single_getactivegroup(0x7ffc3dd122a0, 0x18, ...) kernel cache redo
| | | | > kspGetInstType(0x1, 0x1, ...) kernel service parameter validate inst
| | | | > kcrfw_slave_phase_enter(0x1, 0x78, ...) kernel cache redo file write ??
| | | | > kcscu8(0x60016290, 0x7ffc3dd11858, ...) kernel cache scn man. read current SCN
| | | | > kcrfw_update_blk_list(0x77eced90, 0x7ffc3dd11858, ...) kernel cache redo file wr
| | | | > kslgetl(0x6004b780, 0x1, ...) kernel service lock man. get latch
| | | | > kslgetl(0x76fe0c10, 0x1, ...)
| | | | > kslfre(0x76fe0c10, 0x1, ...) kernel service lock man. free latch
| | | | > kslfre(0x6004b780, 0x1, ...)
| | | | > skgstmGetEpochTs(0x6004b780, 0x1, ...) o/s dependent kernel generic service t
| | | | > kslgetl(0x6004b780, 0x1, ...)
| | | | > ub8_to_kscn_impl(0x78251b, 0x600162c0, ...) convert ub8 to kernel SCN
| | | | > kcsnew_rba(0x7ffc3dd11800, 0x600162c8, ...) kernel cache scn man. write new rba
| | | | > kslfre(0x6004b780, 0x1, ...)
| | | | > krsht_trace(0x200, 0x1, ...) kernel recovery standby/dataguard ??
| | | | > kslpsemf(0x91, 0, ...) kernel service lock man. post semaphore
| | | | > kcrfw_stats_writeinfo_insert(0x7ffc3dd118b8, 0x7ffc3dd11510, ...) kcrfw_stats_wr
| | | | > kcsaj8(0x60016290, 0x7ffc3dd118b8, ...) kernel cache scn management adjust SCN
| | | | > kcrfw_slave_phase_exit(0x1, 0x78, ...) kernel cache redo file write ??
| | | | > kcbbposts(0x1, 0x78, ...) kernel cache buffers databasewriter ??
| | | | > _intel_fast_memcpy(0x769a5730, 0x76fe0af0, ...)
| | | | > kcrfw_slave_group_splitevents(0x769a5730, 0x76fe0af8, ...) kernel cache redo fil
| | | | > kslpslf(0x97, 0x769a5730, ...) kernel service locking post list of fo
| | | | > skgstmGetEpochTs(0x7f2c23b5e6c0, 0x7ae0ee70, ...) o/s dependent kernel generic
```

KSBCTI - 2

```
| | | > kslpslf(0x97, 0x769a5730, ...) kernel service locking p
| | | | > ksl_postm_init(0x7ffc3dd094f0, 0x7ffc3dd11510, ...) kernel service locking p
| | | | > kslget1(0x7a5d7388, 0x1, ...) kernel service locking c
| | | | > ksl_update_post_stats(0x7a619f48, 0x7ffc3dd11510, ...) kernel service locking u
| | | | > kslfre(0x7a5d7388, 0, ...) kernel service locking i
| | | | > ksl_postm_do_posts(0x7ffc3dd094f0, 0, ...) kernel service locking p
```

KSBCTI - 2

```
| | | | > ksl_postm_do_posts(0x7ffc3dd094f0, 0, ...) kernel service locking post
| | | | | > kslpwrp_int(0x7ffc3dd09450, 0x7f2c23b9e9a0, ...) kernel service locking post
| | | | | | > skgpwpost(0x7ffc3dd09350, 0x7f2c23ba3ca0, ...) o/s dependent kernel gener
| | | | | | <> sskgpwpost(0x7ffc3dd09350, 0x7f2c23ba3ca0, ...) o/s dependent system kerne
| | | | | | | > semop@plt(0x38000, 0x7ffc3dd07310, ...)
| | | | | | | < semop+0x000000000000f returns: 0
| | | | | | | < sskgpwpost+0x0000000000009a returns: 0x1
| | | | | | < kslpwrp_int+0x00000000000069 returns: 0x1
| | | | | | > kgecss(0x7f2c23b9e9a0, 0x7f2c1e54b048, ...) kernel generic error ??
| | | | | | < kgecss+0x00000000000035 returns: 0x7f2c23be0f40
| | | | | < ksl_postm_do_posts+0x0000000000144 returns: 0x1
```

LOG WRITER POSTED

```
> ksarcv(0x7ffc3dd12338, 0x7ffc3dd122a0, ...)  
> ksl_exit_main_loop_wait(0x6003ef40, 0x7af972b8, ...)  
> kcrfw redo write driver(0x7ffc3dd122a0, 0x18, ...) <<<<<  


---

  
> ksbcti(0x12b84c74, 0x7f2c1eb804b0, ...)  
> ksarcv(0x7ffc3dd12338, 0x7ffc3dd122a0, ...)  
> dbktFlush(0x7ffc3dd12338, 0x7ffc3dd122a0, ...)  
> sltrgatime64(0x7f2c23b9e9a0, 0x7f2c23b9ebe8, ...)  
> ksl_enter_main_loop_wait(0x1, 0x7a6032e8, ...)  
> ksarcv(0x7ffc3dd12338, 0x7ffc3dd122a0, ...)
```

LOG WRITER - IDLE

Conclusion on the active loop of the log writer:

- The log writer either a) finds new strand entries or b) is posted by a process.
- More work is done, memory buffers used in the shared pool and PGA memory allocated.
- More non shared latches access:
 - The same latches in the idle loop: messages, lgwr LWN SCN, redo allocation. All in willing to wait mode.
 - Plus: redo writing (3x), redo allocation, Consistent RBA, log write info, post/wait queue.
- The shared latches: KTF sga latch, Nologging Standby SCN Time Map cache are omitted when the log writer is posted.
- When log writing IO is done, it's covered by 'log file parallel write'; which includes the FULL IO time.
- Foreground processes are posted via semop system call when not in polling mode.

LOG FILE SYNC TUNING

If individual 'log file sync' wait times gets high:

a) Is the logwriter spending the time on doing IO (log file parallel write)?

Take the actual 'log file parallel write' figures from the same time as your 'log file sync' figures.

b) Does the LGWR show any other significant time spend in wait events at that time?

'LGWR wait for redo copy' or 'latch: redo allocation' are most likely.

c) Does the logwriter spend its time on CPU?

As we have seen, an active log writer is doing more, but nothing really substantial outside things covered by a wait event, *outside of posting processes*.

d) Does the log writer sit in the run queue waiting to become runnable?

Not easy to measure directly. Reasonably easy indirectly; (1 min) load vs. # core's.

The main time spend by an active log writer should be IO time (log file parallel write).

LOG FILE SYNC TUNING

When overall time spend on 'log file sync' is high: commit only when needed. Probably at the end of your transaction.

The change vectors will be flushed to the log buffer without commit. Use commit as the logical switch to make your data changes visible.

Over-committing is a common source of performance problems. In general it shows a lack of understanding how Oracle database mechanics work.

***A workaround* for bad application design: set the database fixed to polling mode.
(alter database set '_use_adaptive_log_file_sync'=polling_only scope=spfile;)**

LINKS

Intel pin tools: <https://software.intel.com/en-us/articles/pin-a-dynamic-binary-instrumentation-tool>

Pinatrace annotate: https://gitlab.com/FritsHoogland/pinatrace_annotate

Oracle functions: <http://orafun.info>