Crash Dump Analysis
TRAPTRACE

Jakub Jermář
Martin Děcký
When DTrace is not enough

- Mostly assembly language code without epilogues and prologues
- When there are no sdtt probes
- When the context is very restricted, such as callback_handler or trap code
- If there is no DTrace, i.e. Solaris 9 and older
When kmdb is not enough

- Set a breakpoint, but the debugger fails when the breakpoint is hit
- Set a breakpoint, the kernel crashes without hitting the breakpoint
- Set a breakpoint, the breakpoint is hit, but the kernel crashes upon continuing
TRAPTRACE

• Low-level tracing of essential system events
• Available in Solaris
  ▪ IA-32/AMD64 and SPARC V9 (sun4u/sun4v)
    - Slightly different implementations
• May be the only analysis aid left when ...
  ▪ ... everything else fails
  ▪ ... other techniques are not applicable
TRAPTRACE (2)

- Compile-time choice
  - Cannot be turned on if not present
  - Cannot be turned off if present
  - Enabled in debug kernels
  - When TRAPTRACE macro defined

- MDB support
  - Present the trace data from a crash dump
    - Requires post-mortem interpretation by a human
Implementation – sun4u

• Trace data stored in a per-CPU kernel circular buffer of records of the `struct trap_trace_record` type

```c
struct trap_trace_record {
    uint16_t        tt_tl;
    uint16_t        tt_tt;
    uintptr_t       tt_tpc;
    uint64_t        tt_tstate;
    uint64_t        tt_tick;
    uintptr_t       tt_sp;
    uintptr_t       tt_tr;
    uintptr_t       tt_f1;
    uintptr_t       tt_f2;
    uintptr_t       tt_f3;
    uintptr_t       tt_f4;
};
```
Implementation – sun4u (2)

- **tt_tl**
  - corresponds to the TL register as it existed in the moment of the event
  - trap level
    - (0) – no trap in progress
    - (1) – a trap in progress
    - (>1) – nested trap in progress
      - depth of nesting
Implementation – sun4u (3)

- **tt_tt**
  - **trap type**
    - **0x0 – 0x1ff**
      - identifies the type of the trap
        - page fault vs. interrupts vs. window trap etc.
    - **>= 0x200**
      - for non-trap events
        - such as TSB-miss / hit
        - passing a trace-point in the code
Implementation – sun4u (4)

- **tt_tpc**
  - corresponds to the TPC register as it existed in the moment of the trap
  - trap PC
    - records the address in code where the event occurred

- **tt_tstate**
  - snapshot of the TSTATE register as it existed in the moment of the trap
    - information about processor state
Implementation – sun4u (5)

- **tt_tick**
  - corresponds to the STICK register as it existed in the moment of the event
  - event timestamp

- **tt_sp**
  - snapshot of the SP register as it existed in the moment of the event
Implementation – sun4u (6)

- **tt_tr**
- **tt_f1 - tt_f4**
  - auxiliary fields used by non-trap records
    - e.g. details about MMU faults, register windows configuration registers
Instrumentation – sun4u

• Spot the difference

> trap_table0+98*20,20/ai
0x1001300:
0x1001300: stx %l0, [%sp + 0x7ff]
...
0x100131c: stx %l7, [%sp + 0x837]
0x1001320: stx %i0, [%sp + 0x83f]
...
0x1001338: stx %fp, [%sp + 0x86f]
0x100133c: stx %i7, [%sp + 0x877]
0x1001340: ba +0x60dc <0x100741c>
0x1001344: rd %pc, %l4
0x1001348: clr %l4
0x100134c: saved
0x1001350: retry

> trap_table0+98*20,20/ai
0x1001300:
0x1001300: stx %l0, [%sp + 0x7ff]
...
0x100131c: stx %l7, [%sp + 0x837]
0x1001320: stx %i0, [%sp + 0x83f]
...
0x1001338: stx %fp, [%sp + 0x86f]
0x100133c: stx %i7, [%sp + 0x877]
0x1001340: saved
0x1001344: retry
0x1001348: illtrap 0x0
0x100134c: illtrap 0x0
0x1001350: illtrap 0x0
Instrumentation – sun4u

• Spot the difference

> trap_table0+98*20, 20/ai
0x1001300:
0x1001300: stx %l0, [%sp + 0x7ff]
...  
0x100131c: stx %l7, [%sp + 0x837]
0x1001320: stx %i0, [%sp + 0x83f]
...  
0x1001338: stx %fp, [%sp + 0x86f]
0x100133c: stx %i7, [%sp + 0x877]
0x1001340: ba +0x60dc <0x100741c> 
0x1001344: rd %pc, %l4
0x1001348: clr %l4
0x100134c: saved
0x1001350: retry

> trap_table0+98*20, 20/ai
0x1001300:
0x1001300: stx %l0, [%sp + 0x7ff]
...  
0x100131c: stx %l7, [%sp + 0x837]
0x1001320: stx %i0, [%sp + 0x83f]
...  
0x1001338: stx %fp, [%sp + 0x86f]
0x100133c: stx %i7, [%sp + 0x877]
0x1001340: saved
0x1001344: retry
0x1001348: illtrap 0x0
0x100134c: illtrap 0x0
0x1001350: illtrap 0x0
Instrumentation – sun4u (2)

- TT_TRACE(label) macro
  - `trace_gen`
  - `trace_win`
  - `trace_tsbmiss`
  - `trace_tsbhit`
Instrumentation – sun4u (3)

- SYSTRAP_TRACE
  - tracing the sys_trap() trace-point
- Directly embedded
  - pil_interrupt()
MDB Support

- MDB can present the TRAPTRACE data collected before crash
- The data can be used to reconstruct events which lead to a crash
- `[cpuid]::ttrace [-x]`
## MDB Support (2)

```plaintext
> ::tttrace

<table>
<thead>
<tr>
<th>CPU</th>
<th>%tick</th>
<th>%tt</th>
<th>%tl</th>
<th>%tpc</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000000000c40ced44</td>
<td>0024</td>
<td>cleanwin</td>
<td>0001 0000000000108e4c0 vsnprintf</td>
</tr>
<tr>
<td>0</td>
<td>0000000000c40ced1f</td>
<td>0268</td>
<td>?</td>
<td>0001 0000000001087704 panicsys+0x120</td>
</tr>
<tr>
<td>0</td>
<td>0000000000c40cecfb</td>
<td>0098</td>
<td>spill-6-norm</td>
<td>0001 00000000010086e4 flush_windows+4</td>
</tr>
<tr>
<td>0</td>
<td>0000000000c40cecf5</td>
<td>0098</td>
<td>spill-6-norm</td>
<td>0001 00000000010086e4 flush_windows+4</td>
</tr>
</tbody>
</table>
```
### MDB Support (3)

```plaintext
> 0::ttrace -x

<table>
<thead>
<tr>
<th>%tick</th>
<th>%tstate</th>
<th>%tt</th>
<th>%tl</th>
<th>%tpc</th>
<th>%sp</th>
</tr>
</thead>
<tbody>
<tr>
<td>TR</td>
<td>F1-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| 0000000000c40ced44 | 00000000000001606 | 0024 | 0001 | 000000000108e4c0 | 0000000000000000 |
| 00000000000009999 | [15, 7030003, 3000e, 0] |
| 0000000000c40ced1f | 00000000000001c0   | 0268 | 0001 | 0000000001087704 | 00000000070002000 |
| 0000000003f575c00 | [fffffffffffffff, 1087708, 3f680010, 0] |
| 0000000000c40cecfb | 0000009900001603 | 0098 | 0001 | 0000000001086e4 | 0000000000180d5d1 |
| 00000000000009999 | [15, 2050001, 3000e, 1087be4] |
| 0000000000c40cecf5 | 0000009900001603 | 0098 | 0001 | 0000000001086e4 | 0000000000180d681 |
| 00000000000009999 | [15, 1040002, 3000e, 102ae9c] |
```