

Frysk

Tracing and Debugging from Userland

Tracing Workshop, Montreal Canada 2008

Andrew Cagney

Red Hat Canada Inc

Overview

- Brief demo
- Ptrace vs Utracer
 - Pushing code into the kernel
- Back tracing
 - The costs of better quality data
- Variables
 - The search for better quality data

Monitoring View

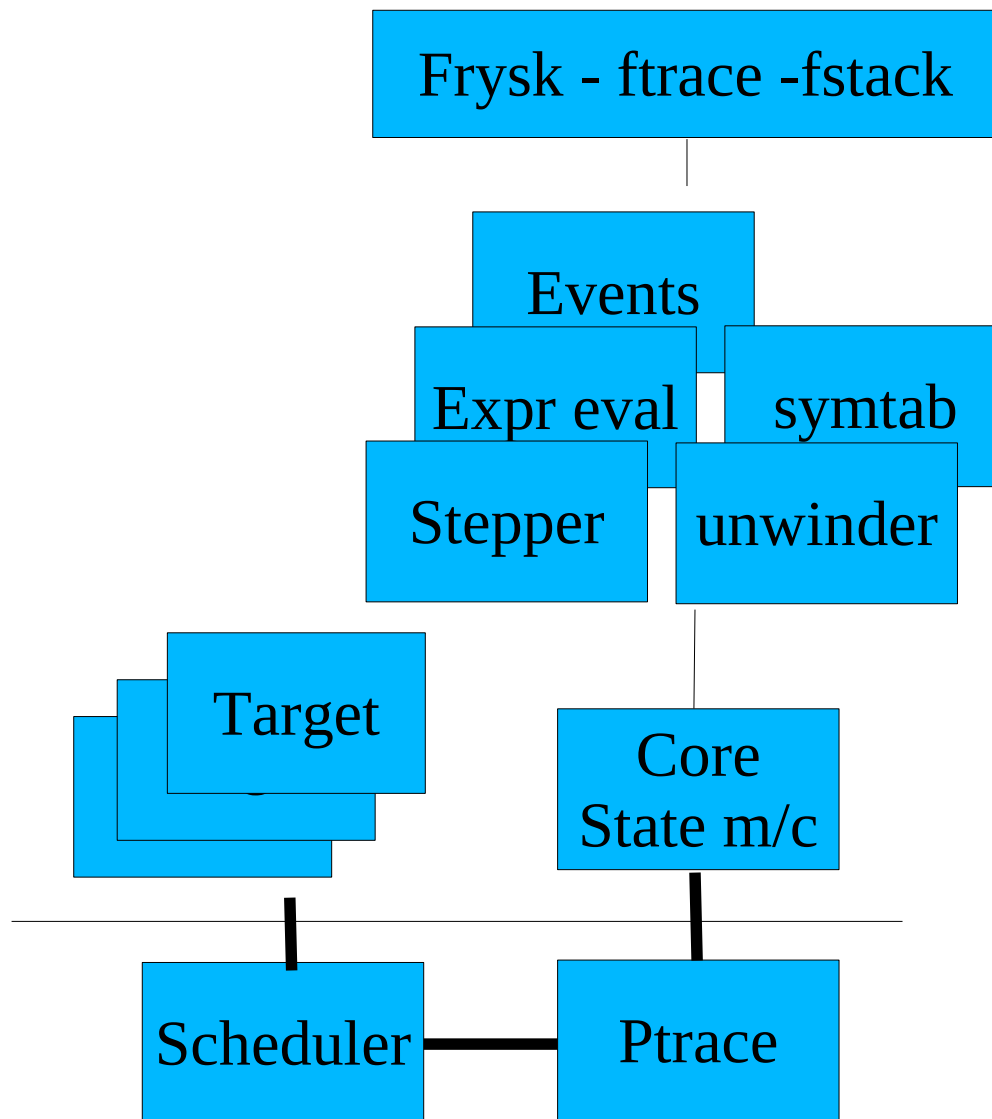
Source Code View

Monitoring vs Debugging

- Non-interactive
- Forensic
- Light weight
- System wide
- Interactive
- Interrogative
- Heavy weight
- Narrow focus

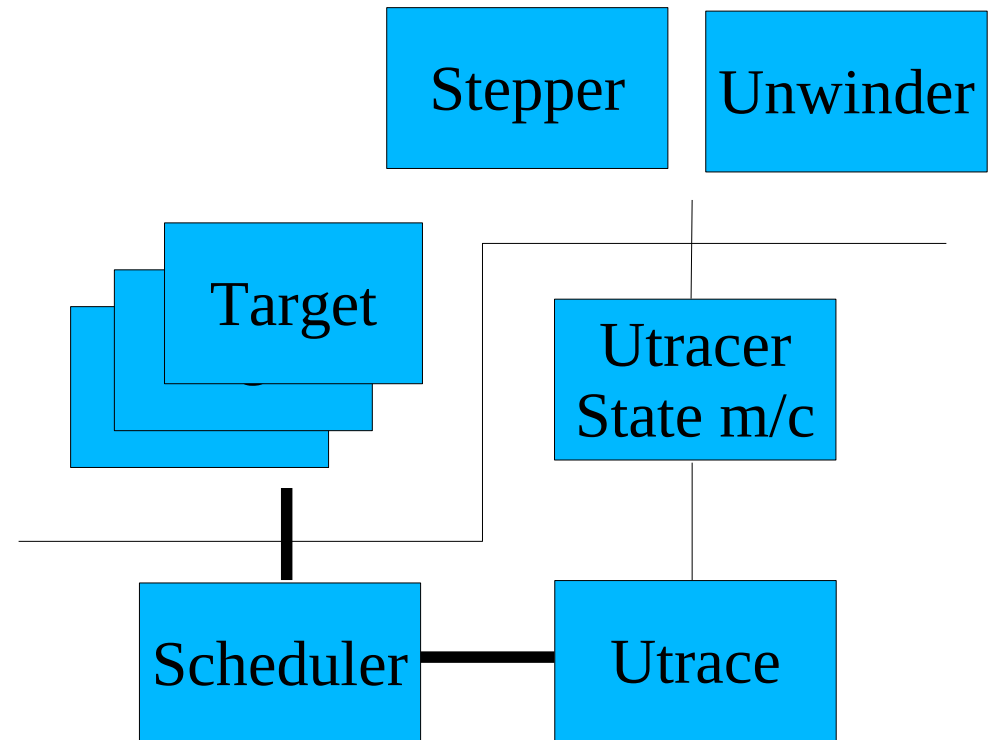
Current Architecture

- Ptrace
 - Single threaded
 - Poor semantics
 - Non-deterministic
 - High overhead
- /proc
 - Different information
 - Same limitations



With Utracer Interface

- Multi-threaded
- Less context switching
 - Far lower overhead
- Friendly to SystemTAP
- Migrate more functionality e.g., stepper/unwinder down



Unwinding

- Modern calling conventions require unwind tables
- `.eh_frame` (in memory)
- `.debug_info` (separate)
- Quality matters
- Size matters

Unwind Info Size

`.eh_frame` vs `.debug_info`

- Part of text segment
- Not default for C
- Required by C++
- Not always in memory (perturbs monitoring)
- Increases text size
~5-10% for C
- In unstripped objects
- Not default - `-g`
- Need to keep unstripped objects
- Dedicated download to kernel for probe use

Unwind Quality

Synchronous vs Asynchronous

- Synchronous

- Non-preemptive
- Unwind points known ahead of time
- System calls

0 `sys_write()`

1 `<kernel-switch>`

2 `write()` [inline]

3 `printf()`

4 `timer_handler()`

5 `<signal-trampoline>`

6 `main()`

Synchronous

- Asynchronous

- Preemptive
- Signals/interrupts
- context switch

Asynchronous

Synchronous vs Asynchronous Unwinding #2

- Can't unwind past “interrupts”
- Default for `.eh_frame`
- “Complete” back trace
- Default for `.debug_info`
- Required for stepping
- Required for C++
- <10% bigger
- `-fasynchronous-unwind-tables`

Variable Information

- Sometimes its wrong
- Sometimes its missing

```
#7 0x0805dc47 in main(int argc = 1,char * * argv = 0xbfb5484,  
char * * env = 0xbfb548c) /usr/src/debug/bash-3.2/shell.c#1161  
{  
  int i = < optimized out > line#335  
  int code = < optimized out > line#336  
  int old_errexit_flag = < optimized out > line#336  
  int saverst = < optimized out > line#338  
  int locally_skip_execution = < optimized out > line#340  
  int arg_index = < optimized out > line#341  
  int top_level_arg_index = < optimized out > line#341  
}
```

Going Forward

- Pushing functionality into the kernel
- Raise quality of unwind and debug information

Questions?

- <http://sourceware.org/frysk>
- <mailto:frysk@sourceware.org>
- <irc://gimp.net/frysk>
- Blog <http://planet.fryskproject.org>
- Test farm <http://build.alchar.org/~aedil/rep/>
- Call Wednesday's @14:30 UTC