Road ahead for Uprobes

plans and features in pipeline.

Srikar Dronamraju – srikar@linux.vnet.ibm.com
IBM Linux Technology Center
Bangalore India

Aug 2012

State of Uprobes

- Merged in 3.5 kernel.
- Lots of contributions from people by way of reviews, fixes, testing.
- Special thanks to:
  - Jim Keniston, Peter Zijlstra, Oleg Nesterov, Ingo Molnar for significant reviews / contributions/ suggestions / fixes.
- Still lot more work ahead.

Return probes

- What are return probes?
- Two approaches to implement return probes
  - Trampoline approach
    - Similar to kretprobes
  - Program/instruction analysis approach
    - May not work for tailcall optimization

- Trampoline approach already maintained by Anton Arapov from Red Hat
  - Currently hosted at

https://github.com/arapov/linux-aa/commits/uretprobes

Prefiltering

- Currently only postfiltering; i.e filter on output.
- All instances of traced app/library take breakpoint hit.

- With prefiltering:
  - Syscall support.
  - Allow non-root users to use uprobes.
  - Allow non-root users to trace their programs only without effecting programs run by other users.

Prefiltering continued.

- Issues in implementing Prefiltering:
  - Threads sharing same ->mm may not be part of primary thread group.
  - How to walk a list of threads that refer to a given mm?
    - Add a list_head to mm
    - Walk thro do_each_thread; while_each_thread;
    - Depend on mm->owner
  - Approach most likely to be accepted is the one suggested by Oleg -- move task->mm to signal_struct https://lkml.org/lkml/2011/6/16/470

Uprobes Syscall

- Who needs it? Perf probe, Gdb and any other debuggers.
- What options should the syscall support?
  - Should we allow stopping a thread so that the tracee can view it?
  - Can we use existing ptrace syscall()?

    or
  - Should we have a new syscall for inserting breakpoints?

Fixmaps

- Cool idea from Oleg Nesterov
  - Area common to all processes address space
  - Having one slot per cpu
  - Use the per cpu slot to singlestep
- Advantages of using fixmaps:
  - No need for xol_vma thats mapped for every traced process
  - No per process pages
  - Efficient and low overhead
- Issues:
  - 32 bit apps over 64 bit kernel?
  - other arch support

Perf probe improvements

- (Statically defined tracepoints aka SDT = user space tracepoints.)
- Already works with SystemTap.
- Can take advantage of DTrace style markers that are already present
- Needed Perf probe support for statically defined tracepoints
  - What should be the role of perf?
- Location/source file based tracing.
- Perf probe gets symbol info wrong if debuginfo is separated from the program

In the works

- Simultaneous ptrace/uprobes tracing
  - Uprobes and ptrace cannot work together.
  - Work already on by Oleg Nesterov and Sebastian Andrzej Siewior.
  - Figure out if the TF bit was already set.
- Powerpc in review.
- Interest in Arm port / S390 port.

Global breakpoints?

- Proposed by Sebastian Andrzej Siewior
- Apply the same breakpoint to multiple processes.
- Uprobes context:
  - the program that hits the breakpoint is held for inspection.
- Should we do this with syscall + task_work_add()?

Legal Statement

- This work represents the view of the authors and does not necessarily represent the view of IBM.
- IBM is a registered trademark of International Business Machines Corporation in the United States and/or other countries.
- Linux is a registered trademark of Linus Torvalds.
- Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Other company, product, and service names may be trademarks or service marks of others.

Thank you