Linux Trace Tools

~Leaving the trace behind

Presented by:
Tarun Sharma
Sharath Koday

Under the guidance of: Prof. Yann Hang Lee
Tracing
- Need for tracing
- Tracing evolution
- Tracepoints
- Why LTTng?
- Antomy
- Data flow
- Understanding complexities of the kernel
- Debugging: Probing into system crashes, page faults
- Potential for module modification/kernel upgrade
- Optimization based on event time recorded, task redundancy

Need for tracing
Tracing Timeline

- 1995: LTT
- 2005: LTTng
- 2008: Ftrace
- 2009: LTTng 2.0
- 2012: Perf

Tracing Timeline
• Static probe points strategically located inside the kernel code

• Register/unregister with tracepoints via callback mechanism

• Can be used to profile, debug and understand kernel behavior

• Ftrace, LTTng, Dtrace provide framework for using probes
Why Use LTTng? *Printf(k) not enough!*

- Low performance impact (Active or inactive)
- Highly optimized user space static instrumentation
- Accurate timestamp
- Dynamic tracing now using kprobes
- “Non root” members can now perform user space tracing
- Attaches context information to events (pid, tid, PMU counter)
Data flow
→ Structure of the tool
→ Creating sessions, deploying probes..
→ Trace interpretation, the eclipse way!
→ Performance showdown: LTTng, Dtrace, SystemTap
→ Conclusion
Tracing: Recording Kernel Trace

koday@ubuntu:/home$ lttng create my_session
Session my_session created.
Traces will be written in /home/koday/lttng-traces/my_session-20121202-125358
koday@ubuntu:/home$ lttng enable-event sched_switch -k
kernel event sched_switch created in channel channel0
koday@ubuntu:/home$ lttng start
Tracing started for session my_session
koday@ubuntu:/home$ lttng stop
Tracing stopped for session my_session
koday@ubuntu:/home$
Tracing: Recording Kernel Trace

Tracing: Interpreting trace using Eclipse
Approx Time by event 8 Threads

- LTTng
- Dtrace
- SystemTap

Tracing Benchmark
Approx Time by event 1Thread

- LTTng
- Dtrace
- SystemTap

Tracing Benchmark
→ Justifies as the successor of LTT with efficient, faster tracing

→ Re-defines performance benchmarks thus promoting competition

→ Active user space support providing options for trace translation

→ Platform independent results in global reach. Progressive understanding and development of Linux kernel
- http://lttng.org/
- kernel.org/doc/html/docs/tracepoint.html
- http://www.eclipse.org/linuxtools/projectPages/lttng/
THANK YOU!