Re: [PATCH 0/11] LTTng−core (basic tracing infrastructure) 0.5.108

Source: http://linux.derkeiler.com/Mailing−Lists/Kernel/2006−09/msg04492.html

- *From*: Mathieu Desnoyers <mathieu.desnoyers@xxxxxxxxxx>
- *Date*: Sat, 16 Sep 2006 13:24:19 −0400

* Jes Sorensen (jes@xxxxxxx) wrote:

> Mathieu Desnoyers wrote:
>
> > Please Ingo, stop repeating false argument without taking in account people's
> > corrections :
> >
> > * Ingo Molnar (mingo@xxxxxxx) wrote:
> >
> > > sorry, but i disagree. There _is_ a solution that is superior in every
> > > aspect: kprobes + SystemTap. (or any other equivalent dynamic tracer)
> >
> > I am sorry to have to repeat myself, but this is not true for heavy loads.
>
> Alan pointed out earlier in the thread that the actual kprobe is noise
> in this context, and I have seen similar issues on real workloads. Yes
> kprobes are probably a little higher overhead in real life, but you have
> to way that up against the rest of the system load.
>
> If you want to prove people wrong, I suggest you do some real life
> implementation and measure some real workloads with a predefined set of
> tracepoints implemented using kprobes and LTT and show us that the
> benchmark of the user application suffers in a way that can actually be
> measured. Argueing that a syscall takes an extra 50 instructions
> because it's traced using kprobes rather than LTT doesn't mean it
> actually has any real impact.
>
> "The 'kprobes' are too high overhead that makes them unusable" is one of
> these classic myths that the static tracepoint advocates so far have
> only been backing up with rhetoric. Give us some hard evidence or stop
> repeating this argument please. Just because something is repeated
> constantly doesn't transform it into truth.

Hi,

Here we go. I made a test that we can consider a lower bound for kprobes impact.
Two tests per run.

Simulation of high speed network traffic :

time ping −f localhost

First run : without any tracing activated, LTTng probes compiled in :

39457 packets received in 2.021 seconds : 19523.50 packets/s
142672 packets received in 7.237 seconds : 19714.24 packets/s

Second run : LTTng tracing activated (traces system calls, interrupts and
packet in/out...) :

93051 packets received in 7.395 seconds : 12582.96 packets/s
121585 packets received in 9.703 seconds : 12530.66 packets/s

Third run : same LTTng instrumentation, with a kprobe handler triggered by each
event traced.

56643 packets received in 11.152 seconds : 5079.17 packets/s
50150 packets received in 9.593 seconds : 5227.77 packets/s

The bottom line is :

LTTng impact on the studied phenomenon : 35% slower

LTTng+kprobes impact on the studied phenomenon : 73% slower

Therefore, I conclude that on this type of high event rate workload, kprobes
doubles the tracer impact on the system.

Mathieu

OpenPGP public key: http://krystal.dyndns.org:8080/key/compudj.gpg
Key fingerprint: 8CD5 52C3 8E3C 4140 715F BA06 3F25 A8FE 3BAE 9A68
−