Subject: Re: [PATCH 2/7] event: don't divide events if it has field period
Posted by Peter Zijlstra on Wed, 09 Nov 2011 11:55:40 GMT
View Forum Message <> Reply to Message

On Mon, 2011-11-07 at 15:54 +0300, Andrew Vagin wrote:
> This patch solves the following problem:
>
> Now some samples may be lost due to throttling. The number of samples is
> restricted by sysctl_perf_event_sample_rate/HZ. A trace event is
> divided on some samples according to event's period. I don't sure, that
> we should generate more than one sample on each trace event. I think the
> better way to use SAMPLE_PERIOD.

It would be yes, but this code predates that, also it needs to work even
if the user doesn't provide SAMPLE_PERIOD.

> E.g.: I want to trace when a process sleeps. I created a process, which
> sleeps for 1ms and for 4ms. perf got 100 events in both cases.
>
> swapper   0 [000] 1141.371830: sched_stat_sleep: comm=foo pid=1801 delay=1386750 [ns]
> swapper   0 [000] 1141.369444: sched_stat_sleep: comm=foo pid=1801 delay=4499585 [ns]
>
> In the first case a kernel want to send 4499585 events and
> in the second case it wants to send 1386750 events.
> perf-reports shows that process sleeps in both places equal time. It's
> bug.
>
> With this patch kernel generates one event on each "sleep" and the time
> slice is saved in the field "period". Perf know how handle it.

Yeah, looks about right, would be awesome if we could strip some
branches out, but nothing obvious comes to mind.

> Signed-off-by: Andrew Vagin <avagin@openvz.org>
> ---
> kernel/events/core.c | 7 ++++++-
> 1 files changed, 6 insertions(+), 1 deletions(-)
>
> diff --git a/kernel/events/core.c b/kernel/events/core.c
> index 12a0287..298702d 100644
> --- a/kernel/events/core.c
> +++ b/kernel/events/core.c
> @@ -4737,7 +4737,6 @@ static void perf_swevent_overflow(struct perf_event *event, u64
overflow,
>   struct hw_perf_event *hwc = &event->hw;
>   int throttle = 0;
>
> - data->period = event->hw.last_period;

>   if (!overflow)
>   overflow = perf_swevent_set_period(event);
>
> @@ -4771,6 +4770,12 @@ static void perf_swevent_event(struct perf_event *event, u64 nr,
>   if (!is_sampling_event(event))
>   return;
>
> + if (event->attr.sample_type & PERF_SAMPLE_PERIOD && !event->attr.freq) {
> + data->period = nr;
> + return perf_swevent_overflow(event, 1, data, regs);
> + } else
> + data->period = event->hw.last_period;
> +
>   if (nr == 1 && hwc->sample_period == 1 && !event->attr.freq)
>   return perf_swevent_overflow(event, 1, data, regs);
>
