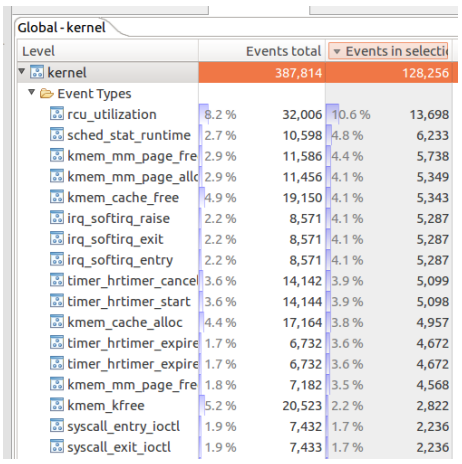
Assigment Report

Isaac Souza

April 25, 2017

This document is a report of a tracing analysis that I have made in order to demonstrate my familiarity with tracing. I used a code that I have written to calculate the approximate value of π using the Archimedes' Method. First, I used a serial version of the program written in C language. Setting the number of iterations to $100,000,000$, I executed the program while using *lttng* to trace the system kernel. Then, I used *trace-compass* to measure the elapsed time (around 2 seconds) and how it interacted with the system kernel. Then, using OpenMP, I made the main loop of the program parallel. Again, using *lttng* and *trace-compass*, I executed the parallelized version of the program and found out that the performance was worst then the serial version (around 4 seconds), different from the expected behavior. In order to find the problem, I imported the two traces on *trace-compass*. Inspecting the parallelized version trace, I saw that each thread on the parallelized version was taking around 4 seconds to complete its job, showing that the problem was with my parallelization.

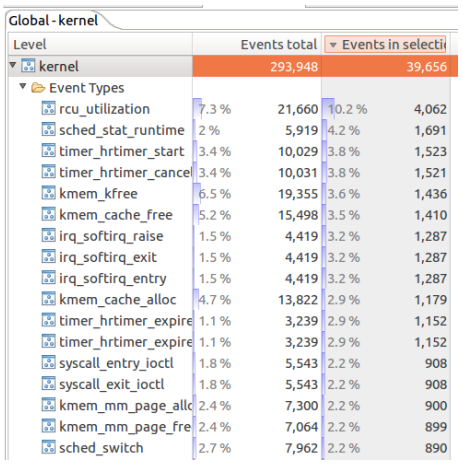
After some quick research and trace analysis, I found out that the problem with my code resided in the fact that each thread was calling the `drand48()` function which is not thread-safe, since it can make the process become locked until another thread finish reading the RNG state data memory address. The figure 1a shows the event count on the time period of the program execution. To address the problem, I used the `srand48_r()` and `drand48_r()` functions instead of the former ones. As a result, this second version of the parallelized program ran faster (around 1 second) then the serial one, as expected. The figure 1b shows the event count for this second version of the code. Comparing the two results on *trace-compass*, I saw that, on the first version, the percentage of events of the type `kmem_page_alloc_zone_locked` (fourth on its list) is bigger than the second one, demonstrating that each thread stays more time locked.



Global - kernel

Level	Events total		Events in selecti	
kernel	387,814		128,256	
Event Types				
rcu_utilization	8.2 %	32,006	10.6 %	13,698
sched_stat_runtime	2.7 %	10,598	4.8 %	6,233
kmem_mm_page_fre	2.9 %	11,586	4.4 %	5,738
kmem_mm_page_allc	2.9 %	11,456	4.1 %	5,349
kmem_cache_free	4.9 %	19,150	4.1 %	5,343
irq_softirq_raise	2.2 %	8,571	4.1 %	5,287
irq_softirq_exit	2.2 %	8,571	4.1 %	5,287
irq_softirq_entry	2.2 %	8,571	4.1 %	5,287
timer_hrtimer_cancel	3.6 %	14,142	3.9 %	5,099
timer_hrtimer_start	3.6 %	14,144	3.9 %	5,098
kmem_cache_alloc	4.4 %	17,164	3.8 %	4,957
timer_hrtimer_expire	1.7 %	6,732	3.6 %	4,672
timer_hrtimer_expire	1.7 %	6,732	3.6 %	4,672
kmem_mm_page_fre	1.8 %	7,182	3.5 %	4,568
kmem_kfree	5.2 %	20,523	2.2 %	2,822
syscall_entry_ioctl	1.9 %	7,432	1.7 %	2,236
syscall_exit_ioctl	1.9 %	7,433	1.7 %	2,236

(a) First version of the program.



Global - kernel

Level	Events total		Events in selecti	
kernel	293,948		39,656	
Event Types				
rcu_utilization	7.3 %	21,660	10.2 %	4,062
sched_stat_runtime	2 %	5,919	4.2 %	1,691
timer_hrtimer_start	3.4 %	10,029	3.8 %	1,523
timer_hrtimer_cancel	3.4 %	10,031	3.8 %	1,521
kmem_kfree	6.5 %	19,355	3.6 %	1,436
kmem_cache_free	5.2 %	15,498	3.5 %	1,410
irq_softirq_raise	1.5 %	4,419	3.2 %	1,287
irq_softirq_exit	1.5 %	4,419	3.2 %	1,287
irq_softirq_entry	1.5 %	4,419	3.2 %	1,287
kmem_cache_alloc	4.7 %	13,822	2.9 %	1,179
timer_hrtimer_expire	1.1 %	3,239	2.9 %	1,152
timer_hrtimer_expire	1.1 %	3,239	2.9 %	1,152
syscall_entry_ioctl	1.8 %	5,543	2.2 %	908
syscall_exit_ioctl	1.8 %	5,543	2.2 %	908
kmem_mm_page_allc	2.4 %	7,300	2.2 %	900
kmem_mm_page_fre	2.4 %	7,064	2.2 %	899
sched_switch	2.7 %	7,962	2.2 %	890

(b) Second version of the program.

Figure 1: The event count for the two version of the parallelized program.