Benchmarking tool
to compare various tracing solutions

Guillaume Duclos-Cianci

July 6, 2015

1 Objective

Create a simple program that benchmarks the getuid system call to extract the overhead introduced by various tracers.

2 Implementation details

2.1 Timestamps

The benchmark is performed in the following manner. First a timestamp is taken using a call to *clock_gettime*. The value is saved in a timespec table. The system call *getuid* is then performed. This process is repeted over and over such that two consecutive timstamps represent the time necessary to perform *clock_gettime* plus *getuid* plus the loop to next round.

Here is a snippet of the code performing the benchmark.

```
for(ts_iter, i; i < BURST_SIZE; ts_iter++, i++){
    clock_gettime(CLOCK_MONOTONIC_RAW, ts_iter);
    getuid();
}
clock_gettime(CLOCK_MONOTONIC_RAW, ts_iter);
```

Tracers:

1. Lttng
2. perf
3. ftrace
4. system tap

2.2 Lttng

The parameters explored for Lttng are the number of subbuffers, *num-subbuf*, and their size, *subbuf-size*. We also compare standard output writtent to disk and snapshot mode.

2.3 System tap

Various probe contents were tested. In all cases, the scaling was prohibitive.