Real Time Embedded Systems – ECEN 5623

Project Report Extension/Update to the Original One

Summer 2019

Professor Sam Siewert

Prepared by **Poorn Mehta**

Hardware Used: Jetson Nano with Logitech C920s

Topics That are Included in This Extension

1. **CPU Usage Tracing**
2. **Profiling and Tracing Jitter of All Threads** (Execution Times Profiling and Tracing is done, but not included since the jitter analysis covers most of the information)
3. **Monotonic Analysis** (Proof that the online stopwatch was behaving erratically, not the self-developed real-time system)
4. **Rate Monotonic Analysis with Graphs based on Syslog Traces** (Proves that the system is designed as per stated in earlier report, and is highly accurate as well as precise)

Folders Walkthrough (Submitted)

1. Analysis.zip – Contains analysis files (all Graphs, csv files, and xls files)
2. Calibration_Code.zip – Code for performing calibration on 1Hz, on Analog Ticking Clock
3. Final_1Hz_Frames.7z – Archive containing 7201 PPM RGB images captured at 1Hz, of a ticking analog clock
4. Final_10Hz_Frames.7z – Archive containing 12001 PPM RGB images captured at 10Hz, of a online stopwatch
5. Flow_Charts.zip – Contains all of the System Diagrams
6. Project_Code.zip – Contains the actual code that runs on Jetson Nano (along with configuration script), for the Time Lapse Project
7. Remote_Server.zip – Contains the actual code that runs on Remote Linux System, to continuously download the frames sent by Jetson
8. RMA.zip – Files regarding Rate Monotonic Analysis
9. Screenshots.zip – Many useful images, graphs, and screenshots
10. Sysprof.zip – Screenshots of System Profiling
11. Verification_Videos.zip – All 4 Time Lapse Videos

CPU Usage Tracing

1Hz, Without File Transfer

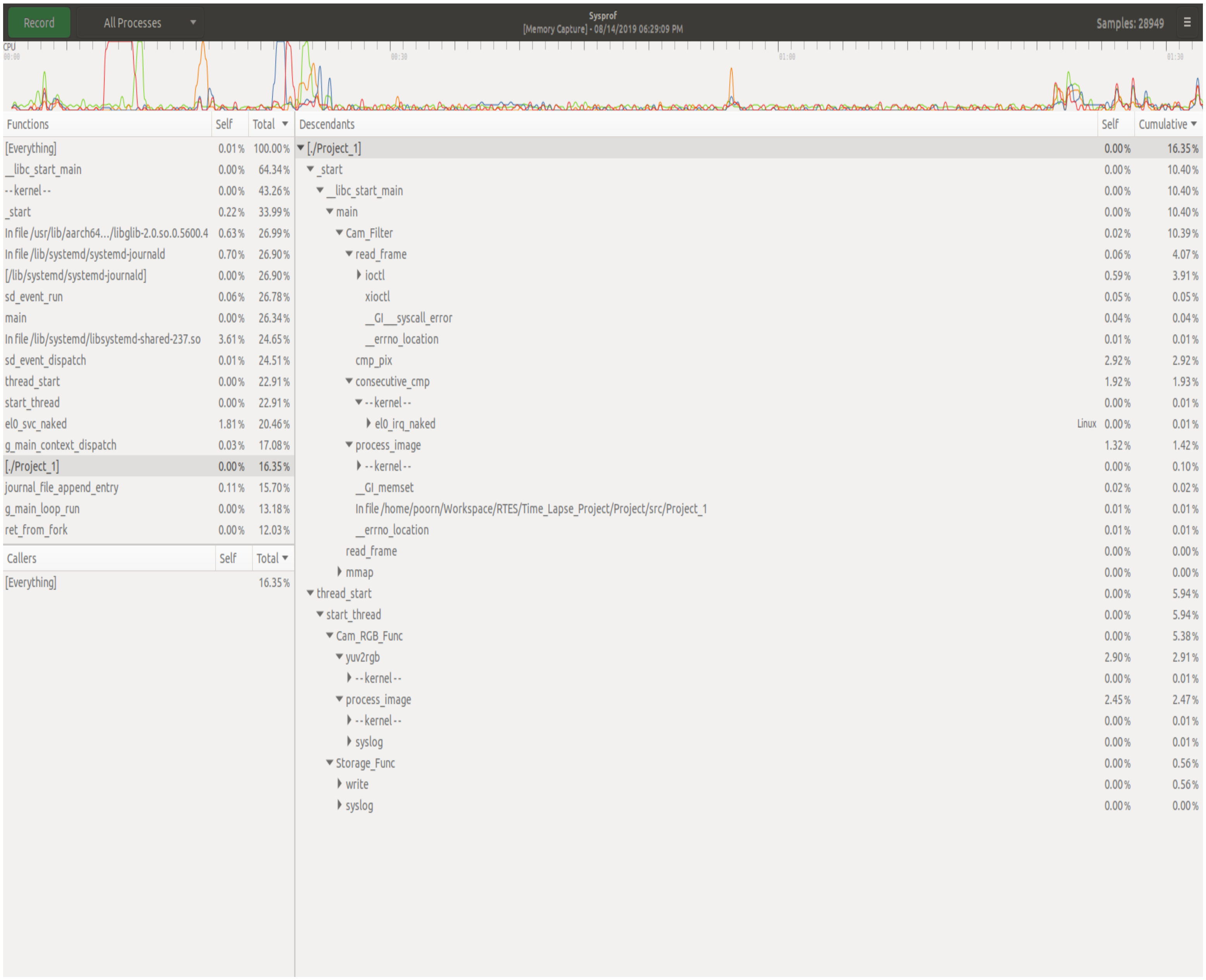
Cumulative Usage: 16.35%

Most Significant Usages:

Cam_Filter – 10.39% (This task captures frames as fast as it can, and compares all consecutive frames as well, in continuous manner before threads are created. Therefore, it is expected that this has a significant CPU loading)

Cam_RGB – 5.38% (This thread converts all pixels of a captured frame, from default YUYV to RGB, and thus – it is also CPU intensive task)

Storage – 0.58% (This thread runs in best effort to write all of the frames on FLASH)



Functions	Self	Total
[Everything]	0.01%	100.00%
__libc_start_main	0.00%	64.34%
--kernel--	0.00%	43.26%
_start	0.22%	33.99%
In file /usr/lib/aarch64.../libglib-2.0.so.0.5600.4	0.63%	26.99%
In file /lib/systemd/systemd-journald	0.70%	26.90%
[/lib/systemd/systemd-journald]	0.00%	26.90%
sd_event_run	0.06%	26.78%
main	0.00%	26.34%
In file /lib/systemd/libsystemd-shared-237.so	3.61%	24.65%
sd_event_dispatch	0.01%	24.51%
thread_start	0.00%	22.91%
start_thread	0.00%	22.91%
el0_svc_naked	1.81%	20.46%
g_main_context_dispatch	0.03%	17.08%
[./Project_1]	0.00%	16.35%
journal_file_append_entry	0.11%	15.70%
g_main_loop_run	0.00%	13.18%
ret_from_fork	0.00%	12.03%

Callers	Self	Total
[Everything]		16.35%

```
Descendants                                                      Self    Cumulative
▾ [./Project_1]                                                  0.00%   16.35%
  ▾ _start                                                       0.00%   10.40%
    ▾ __libc_start_main                                          0.00%   10.40%
      ▾ main                                                     0.00%   10.40%
        ▾ Cam_Filter                                             0.02%   10.39%
          ▾ read_frame                                           0.06%    4.07%
            ▸ ioctl                                              0.59%    3.91%
              xioctl                                             0.05%    0.05%
              __GI___syscall_error                               0.04%    0.04%
              __errno_location                                   0.01%    0.01%
            cmp_pix                                              2.92%    2.92%
          ▾ consecutive_cmp                                      1.92%    1.93%
            ▾ --kernel--                                         0.00%    0.01%
              ▸ el0_irq_naked                             Linux  0.00%    0.01%
          ▾ process_image                                        1.32%    1.42%
            ▸ --kernel--                                         0.00%    0.10%
              __GI_memset                                        0.02%    0.02%
            In file /home/poorn/Workspace/RTES/Time_Lapse_Project/Project/src/Project_1   0.01%   0.01%
              __errno_location                                   0.01%    0.01%
          read_frame                                             0.00%    0.00%
        ▸ mmap                                                   0.00%    0.00%
  ▾ thread_start                                                 0.00%    5.94%
    ▾ start_thread                                               0.00%    5.94%
      ▾ Cam_RGB_Func                                             0.00%    5.38%
        ▾ yuv2rgb                                                2.90%    2.91%
          ▸ --kernel--                                           0.00%    0.01%
        ▾ process_image                                          2.45%    2.47%
          ▸ --kernel--                                           0.00%    0.01%
          ▸ syslog                                               0.00%    0.01%
      ▾ Storage_Func                                             0.00%    0.56%
        ▸ write                                                  0.00%    0.56%
        ▸ syslog                                                 0.00%    0.00%
```

1Hz, With File Transfer

<u>Cumulative Usage</u>: 31.63%

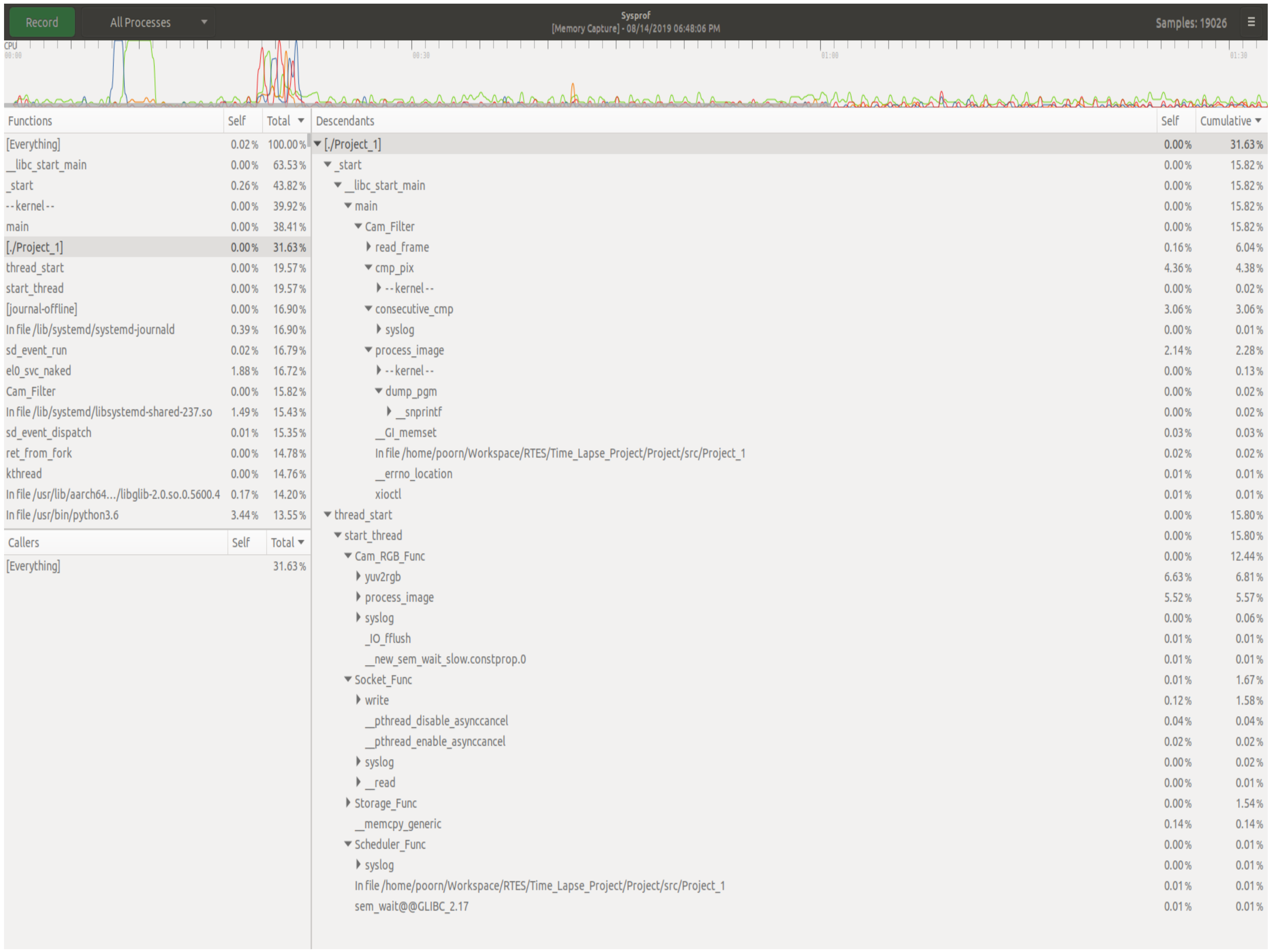
<u>Most Significant Usages</u>:

<u>Cam_Filter</u> – 15.82% (The CPU usage is greater than previous – but the reason is unrelated to the mode. It actually is this way, since it had to ran much longer than previous instance for getting a reliable change in indicator position of analog clock)

<u>Cam_RGB</u> – 12.44% (I am not 100% sure however, I think that at least a part of increase in CPU loading than previous case, is due to more Queue operations, since it has to send all the data to Socket_Queue also)

<u>Socket</u> – 1.67% (This thread transfers 921KB of data per frame, using TCP socket connection, to the Remote Server)

<u>Storage</u> – 1.54% (Not sure how socket has affected increased CPU usage in this)



Functions	Self	Total
[Everything]	0.02%	100.00%
__libc_start_main	0.00%	63.53%
_start	0.26%	43.82%
--kernel--	0.00%	39.92%
main	0.00%	38.41%
[./Project_1]	0.00%	31.63%
thread_start	0.00%	19.57%
start_thread	0.00%	19.57%
[journal-offline]	0.00%	16.90%
In file /lib/systemd/systemd-journald	0.39%	16.90%
sd_event_run	0.02%	16.79%
el0_svc_naked	1.88%	16.72%
Cam_Filter	0.00%	15.82%
In file /lib/systemd/libsystemd-shared-237.so	1.49%	15.43%
sd_event_dispatch	0.01%	15.35%
ret_from_fork	0.00%	14.78%
kthread	0.00%	14.76%
In file /usr/lib/aarch64.../libglib-2.0.so.0.5600.4	0.17%	14.20%
In file /usr/bin/python3.6	3.44%	13.55%

Callers	Self	Total
[Everything]		31.63%

```
Descendants                                                                Self    Cumulative
▾ [./Project_1]                                                            0.00%   31.63%
  ▾ _start                                                                 0.00%   15.82%
    ▾ __libc_start_main                                                    0.00%   15.82%
      ▾ main                                                               0.00%   15.82%
        ▾ Cam_Filter                                                       0.00%   15.82%
          ▸ read_frame                                                     0.16%    6.04%
          ▾ cmp_pix                                                        4.36%    4.38%
            ▸ --kernel--                                                   0.00%    0.02%
          ▾ consecutive_cmp                                                3.06%    3.06%
            ▸ syslog                                                       0.00%    0.01%
          ▾ process_image                                                  2.14%    2.28%
            ▸ --kernel--                                                   0.00%    0.13%
            ▾ dump_pgm                                                     0.00%    0.02%
              ▸ __snprintf                                                 0.00%    0.02%
            __GI_memset                                                    0.03%    0.03%
            In file /home/poorn/Workspace/RTES/Time_Lapse_Project/Project/src/Project_1   0.02%   0.02%
            __errno_location                                               0.01%    0.01%
            xioctl                                                         0.01%    0.01%
  ▾ thread_start                                                           0.00%   15.80%
    ▾ start_thread                                                         0.00%   15.80%
      ▾ Cam_RGB_Func                                                       0.00%   12.44%
        ▸ yuv2rgb                                                          6.63%    6.81%
        ▸ process_image                                                    5.52%    5.57%
        ▸ syslog                                                           0.00%    0.06%
          _IO_fflush                                                       0.01%    0.01%
          __new_sem_wait_slow.constprop.0                                  0.01%    0.01%
      ▾ Socket_Func                                                        0.01%    1.67%
        ▸ write                                                            0.12%    1.58%
          __pthread_disable_asynccancel                                    0.04%    0.04%
          __pthread_enable_asynccancel                                     0.02%    0.02%
        ▸ syslog                                                           0.00%    0.02%
        ▸ __read                                                           0.00%    0.01%
      ▸ Storage_Func                                                       0.00%    1.54%
        __memcpy_generic                                                   0.14%    0.14%
      ▾ Scheduler_Func                                                     0.00%    0.01%
        ▸ syslog                                                           0.00%    0.01%
        In file /home/poorn/Workspace/RTES/Time_Lapse_Project/Project/src/Project_1   0.01%   0.01%
        sem_wait@@GLIBC_2.17                                               0.01%    0.01%
```

10Hz, Without File Transfer

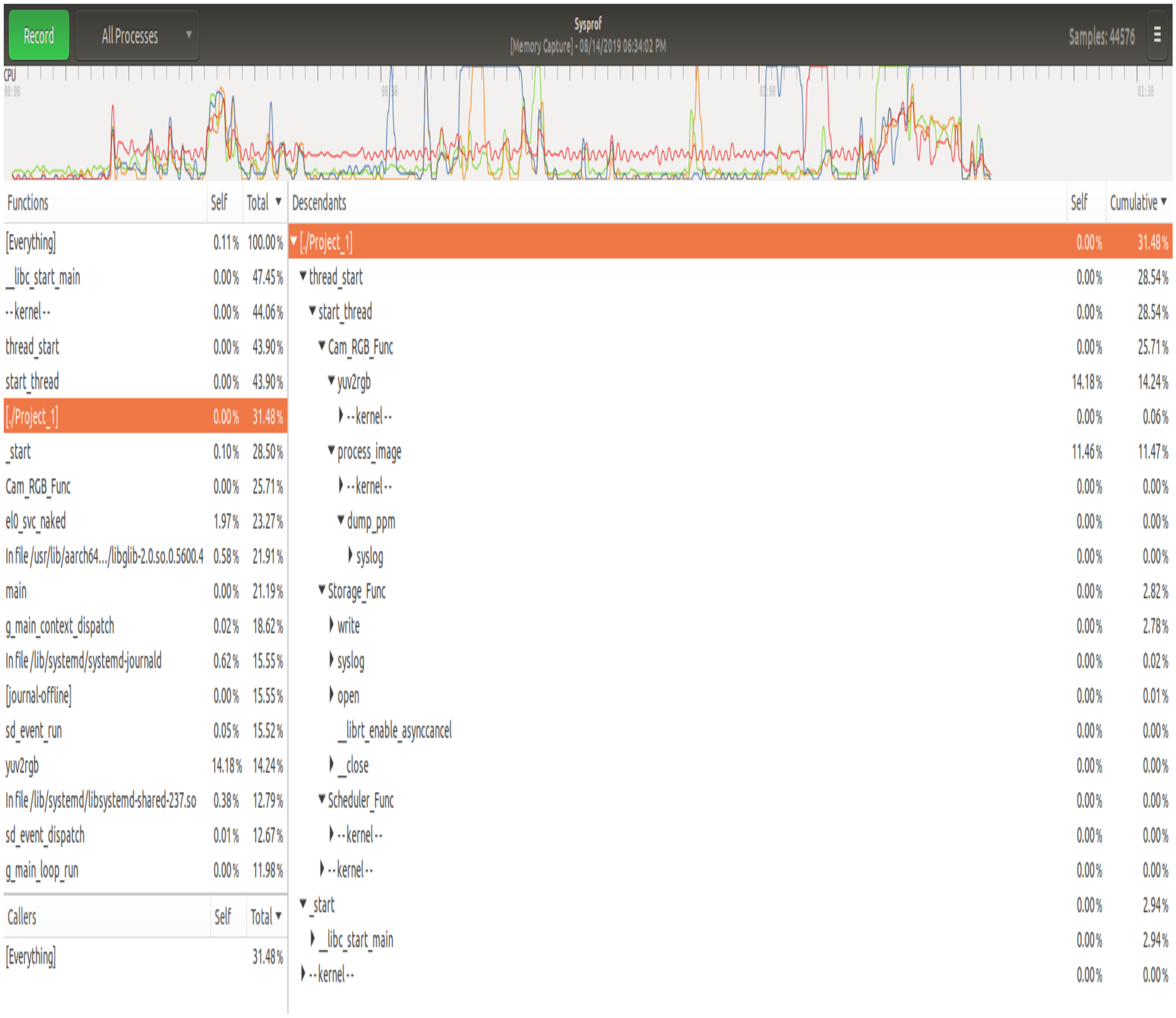
<u>Cumulative Usage</u>: 31.48%

<u>Most Significant Usages</u>:

<u>Cam_RGB</u> – 25.71% (It is obvious that the usage by this thread is much higher, since it has to perform all the previously mentioned operations – 10 times more in the same duration)

<u>Analysis</u> – 2.94% (This function is performed at the end of program, to analyze all of the captured data. There could be thousands of floating-point operations, and therefore it is CPU intensive)

<u>Storage</u> – 2.82% (Higher than 1Hz since it has to write 10 times more data in the same duration)



Functions	Self	Total ▾
[Everything]	0.11%	100.00%
_libc_start_main	0.00%	47.45%
--kernel--	0.00%	44.06%
thread_start	0.00%	43.90%
start_thread	0.00%	43.90%
[./Project_1]	0.00%	31.48%
_start	0.10%	28.50%
Cam_RGB_Func	0.00%	25.71%
el0_svc_naked	1.97%	23.27%
In File /usr/lib/aarch64.../libglib-2.0.so.0.5600.4	0.58%	21.91%
main	0.00%	21.19%
g_main_context_dispatch	0.02%	18.62%
In File /lib/systemd/systemd-journald	0.62%	15.55%
[journal-offline]	0.00%	15.55%
sd_event_run	0.05%	15.52%
yuv2rgb	14.18%	14.24%
In File /lib/systemd/libsystemd-shared-237.so	0.38%	12.79%
sd_event_dispatch	0.01%	12.67%
g_main_loop_run	0.00%	11.98%

Callers	Self	Total ▾
[Everything]		31.48%

Descendants	Self	Cumulative ▾
▾ [./Project_1]	0.00%	31.48%
▾ thread_start	0.00%	28.54%
▾ start_thread	0.00%	28.54%
▾ Cam_RGB_Func	0.00%	25.71%
▾ yuv2rgb	14.18%	14.24%
▸ --kernel--	0.00%	0.06%
▾ process_image	11.46%	11.47%
▸ --kernel--	0.00%	0.00%
▾ dump_ppm	0.00%	0.00%
▸ syslog	0.00%	0.00%
▾ Storage_Func	0.00%	2.82%
▸ write	0.00%	2.78%
▸ syslog	0.00%	0.02%
▸ open	0.00%	0.01%
_librt_enable_asynccancel	0.00%	0.00%
▸ _close	0.00%	0.00%
▾ Scheduler_Func	0.00%	0.00%
▸ --kernel--	0.00%	0.00%
▸ --kernel--	0.00%	0.00%
▾ _start	0.00%	2.94%
▸ _libc_start_main	0.00%	2.94%
▸ --kernel--	0.00%	0.00%

10Hz, With File Transfer

Cumulative Usage: 46.69%

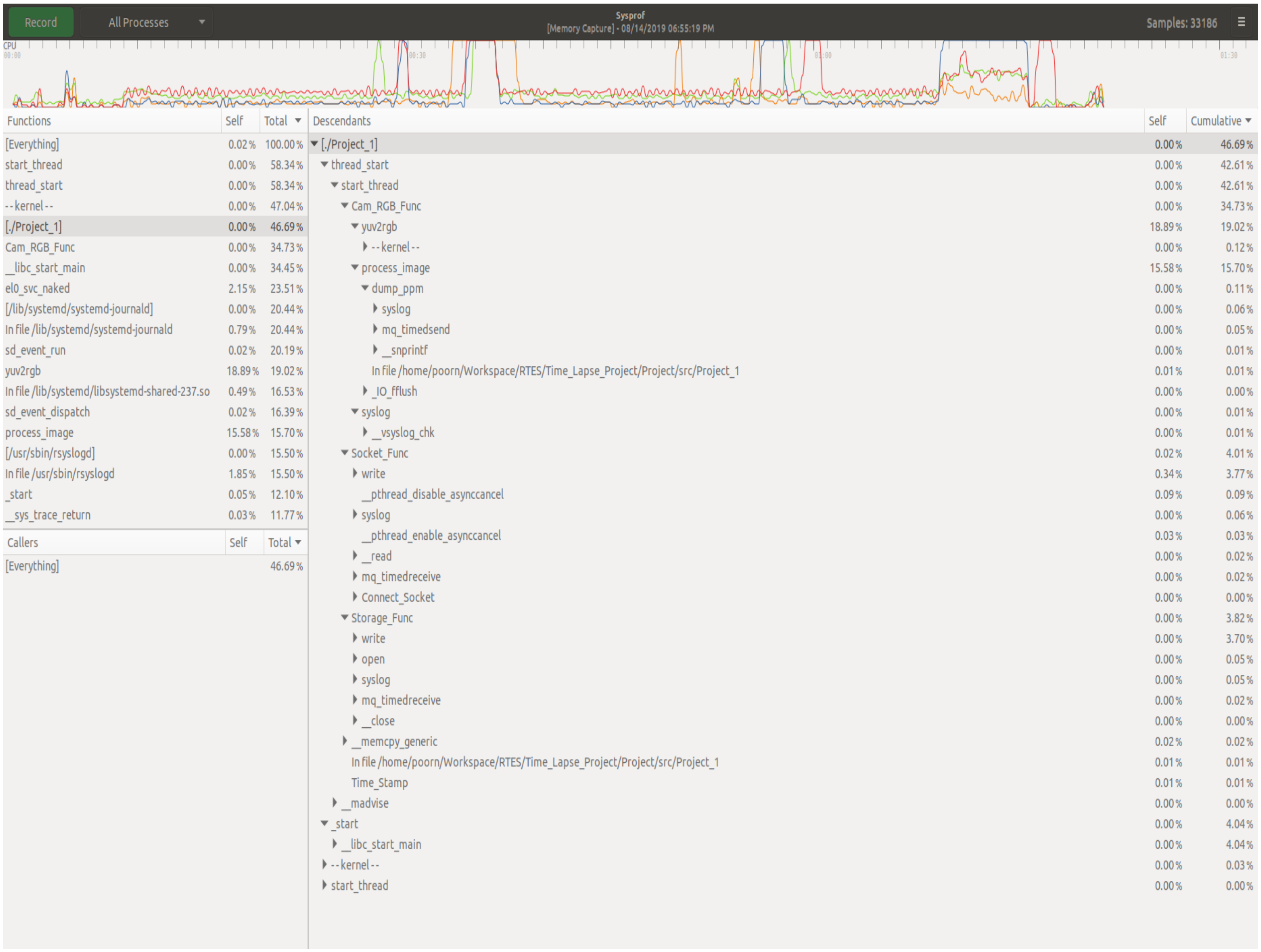
Most Significant Usages:

Cam_RGB – 34.73% (I am not 100% sure however, I think that at least a part of increase in CPU loading than previous case, is due to more Queue operations, since it has to send all the data to Socket_Queue also)

Socket – 4.07% (It is higher than the usage at 1Hz, because it has to send much more data in limited duration, in 10Hz mode)

Analysis – 4.04% (I am not sure why this has increased, since ideally, socket configuration shouldn't affect this particular best effort function)

Storage – 3.82% (Again, I have no idea why enabling socket has increased CPU loading for this thread)



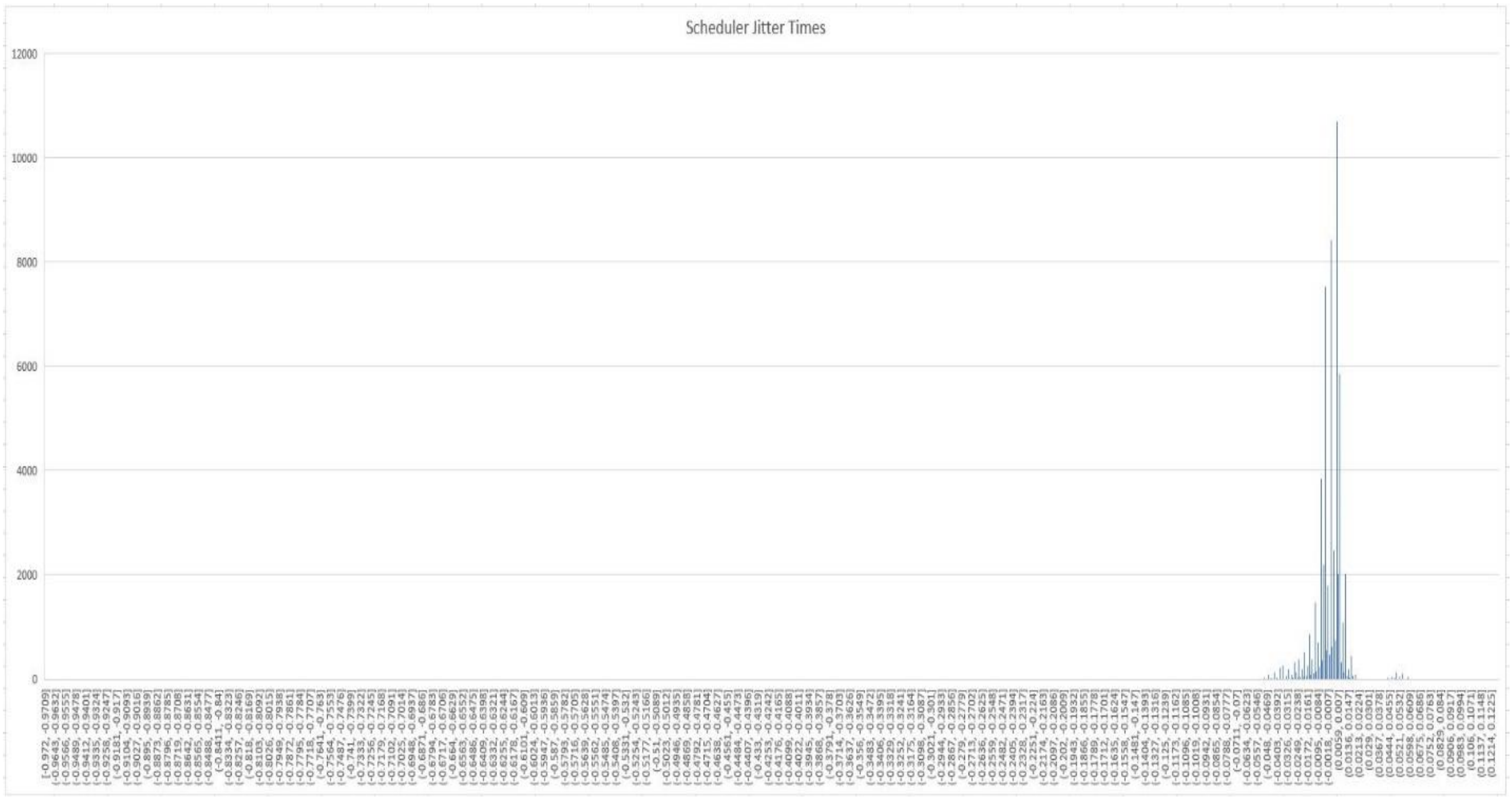
Functions	Self	Total
[Everything]	0.02%	100.00%
start_thread	0.00%	58.34%
thread_start	0.00%	58.34%
- - kernel - -	0.00%	47.04%
[./Project_1]	0.00%	46.69%
Cam_RGB_Func	0.00%	34.73%
__libc_start_main	0.00%	34.45%
el0_svc_naked	2.15%	23.51%
[/lib/systemd/systemd-journald]	0.00%	20.44%
In file /lib/systemd/systemd-journald	0.79%	20.44%
sd_event_run	0.02%	20.19%
yuv2rgb	18.89%	19.02%
In file /lib/systemd/libsystemd-shared-237.so	0.49%	16.53%
sd_event_dispatch	0.02%	16.39%
process_image	15.58%	15.70%
[/usr/sbin/rsyslogd]	0.00%	15.50%
In file /usr/sbin/rsyslogd	1.85%	15.50%
_start	0.05%	12.10%
__sys_trace_return	0.03%	11.77%

Callers	Self	Total
[Everything]		46.69%

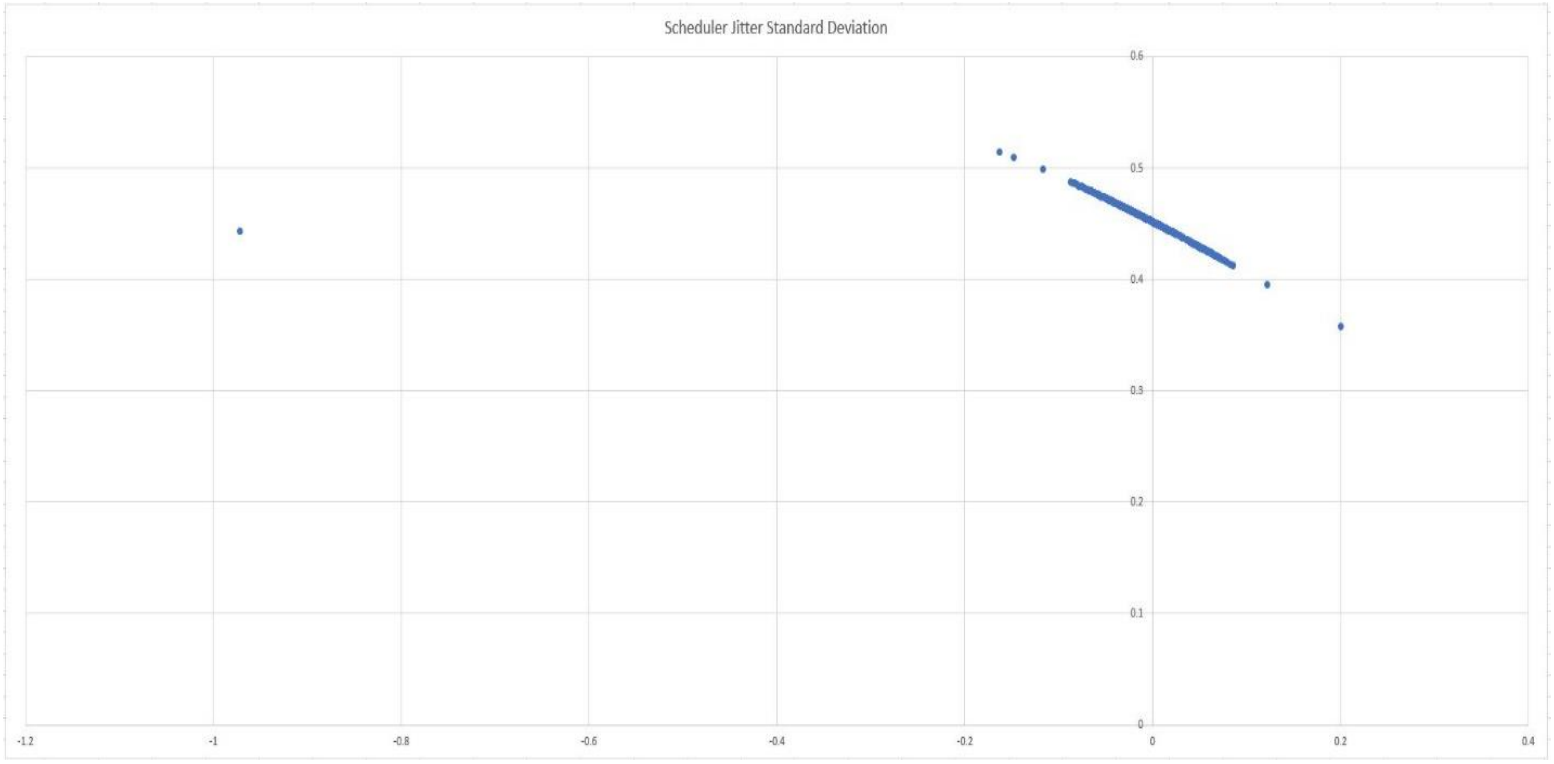
```
Descendants                                                                      Self     Cumulative
▼ [./Project_1]                                                                  0.00%    46.69%
  ▼ thread_start                                                                 0.00%    42.61%
    ▼ start_thread                                                               0.00%    42.61%
      ▼ Cam_RGB_Func                                                             0.00%    34.73%
        ▼ yuv2rgb                                                                18.89%   19.02%
          ▶ - - kernel - -                                                       0.00%     0.12%
        ▼ process_image                                                          15.58%   15.70%
          ▼ dump_ppm                                                             0.00%     0.11%
            ▶ syslog                                                             0.00%     0.06%
            ▶ mq_timedsend                                                       0.00%     0.05%
            ▶ __snprintf                                                         0.00%     0.01%
            In file /home/poorn/Workspace/RTES/Time_Lapse_Project/Project/src/Project_1   0.01%   0.01%
          ▶ _IO_fflush                                                           0.00%     0.00%
        ▼ syslog                                                                 0.00%     0.01%
          ▶ __vsyslog_chk                                                        0.00%     0.01%
      ▼ Socket_Func                                                              0.02%     4.01%
        ▶ write                                                                  0.34%     3.77%
          __pthread_disable_asynccancel                                          0.09%     0.09%
        ▶ syslog                                                                 0.00%     0.06%
          __pthread_enable_asynccancel                                           0.03%     0.03%
        ▶ __read                                                                 0.00%     0.02%
        ▶ mq_timedreceive                                                        0.00%     0.02%
        ▶ Connect_Socket                                                         0.00%     0.00%
      ▼ Storage_Func                                                             0.00%     3.82%
        ▶ write                                                                  0.00%     3.70%
        ▶ open                                                                   0.00%     0.05%
        ▶ syslog                                                                 0.00%     0.05%
        ▶ mq_timedreceive                                                        0.00%     0.02%
        ▶ __close                                                                0.00%     0.00%
      ▶ __memcpy_generic                                                         0.02%     0.02%
      In file /home/poorn/Workspace/RTES/Time_Lapse_Project/Project/src/Project_1   0.01%     0.01%
      Time_Stamp                                                                 0.01%     0.01%
    ▶ __madvise                                                                  0.00%     0.00%
  ▼ _start                                                                       0.00%     4.04%
    ▶ __libc_start_main                                                          0.00%     4.04%
  ▶ - - kernel - -                                                               0.00%     0.03%
  ▶ start_thread                                                                 0.00%     0.00%
```

Profiling and Tracing of all Threads

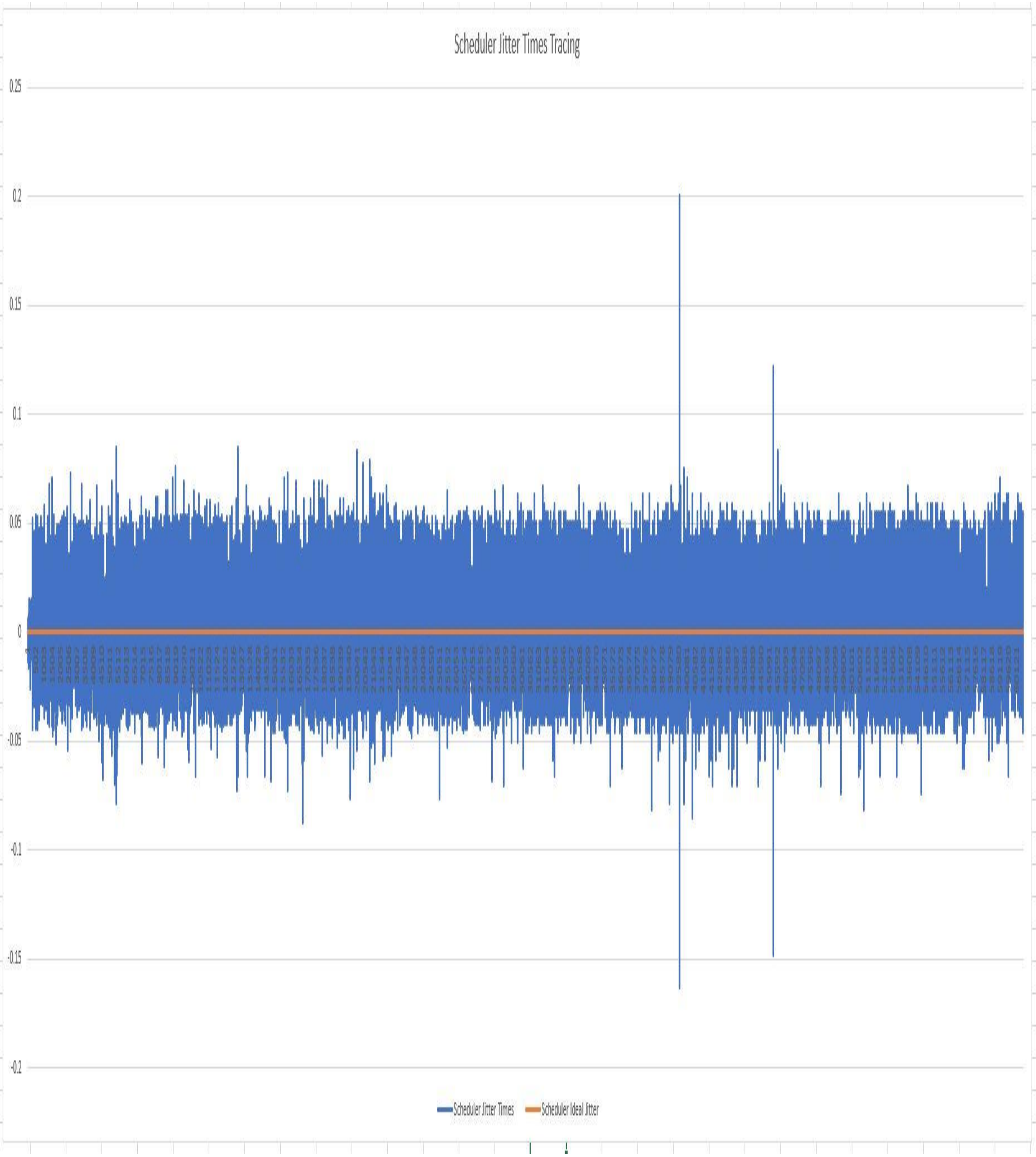
Scheduler Thread Jitter Histogram (Part of Profile)



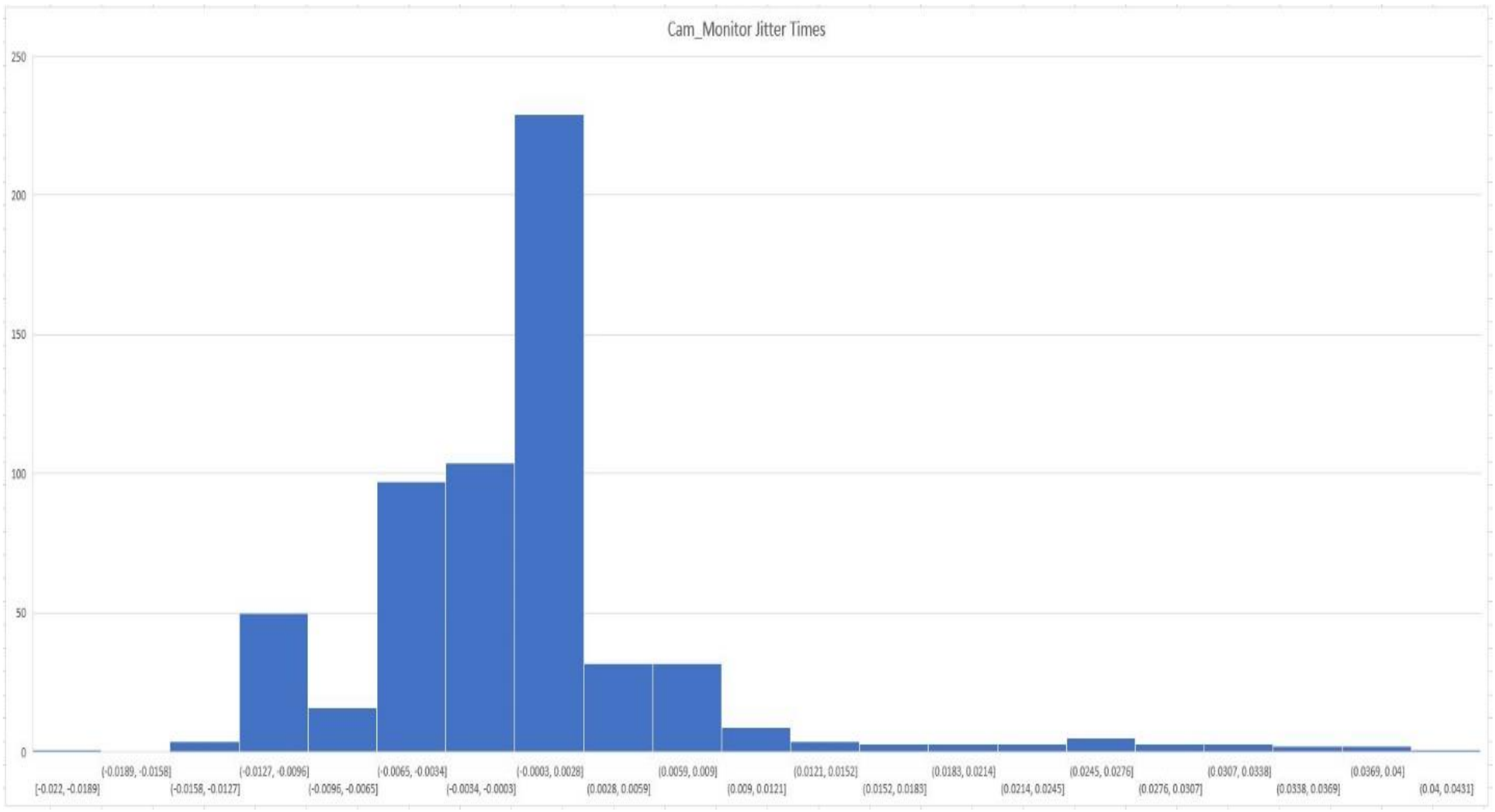
Scheduler Thread Jitter Standard Deviation (Part of Profile)



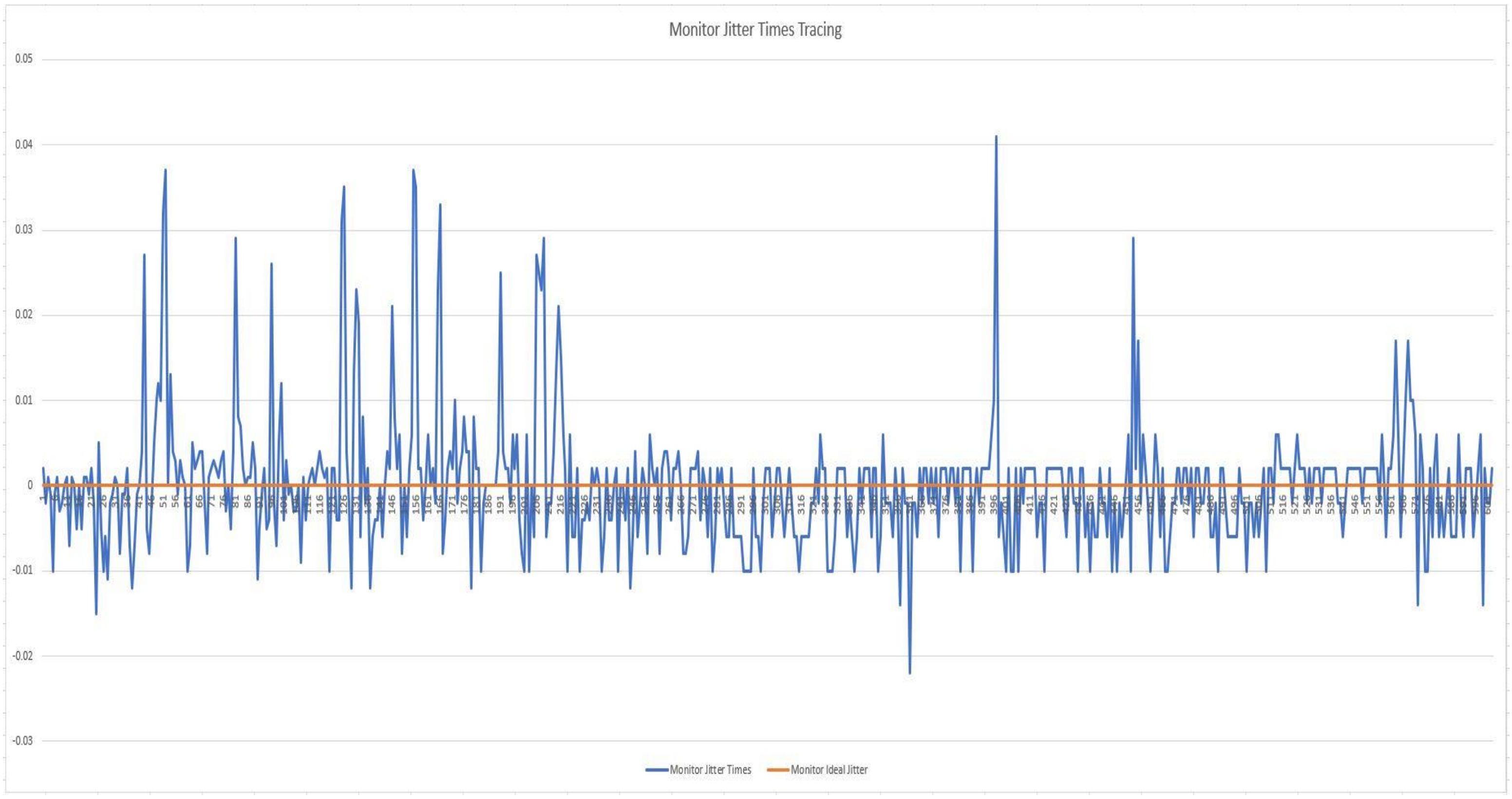
Scheduler Thread Jitter Tracing



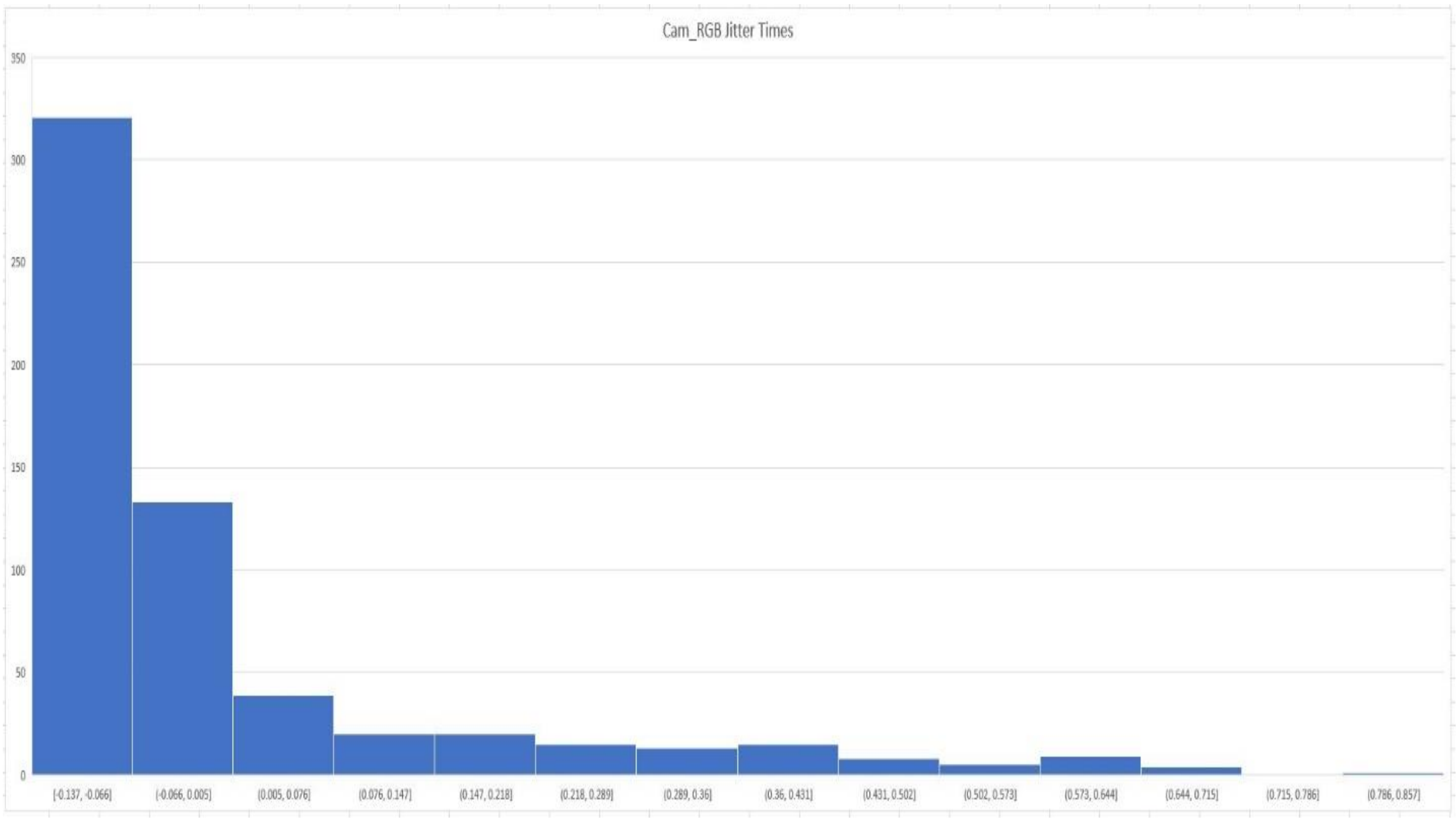
Cam_Monitor Thread Jitter Histogram (Part of Profile)



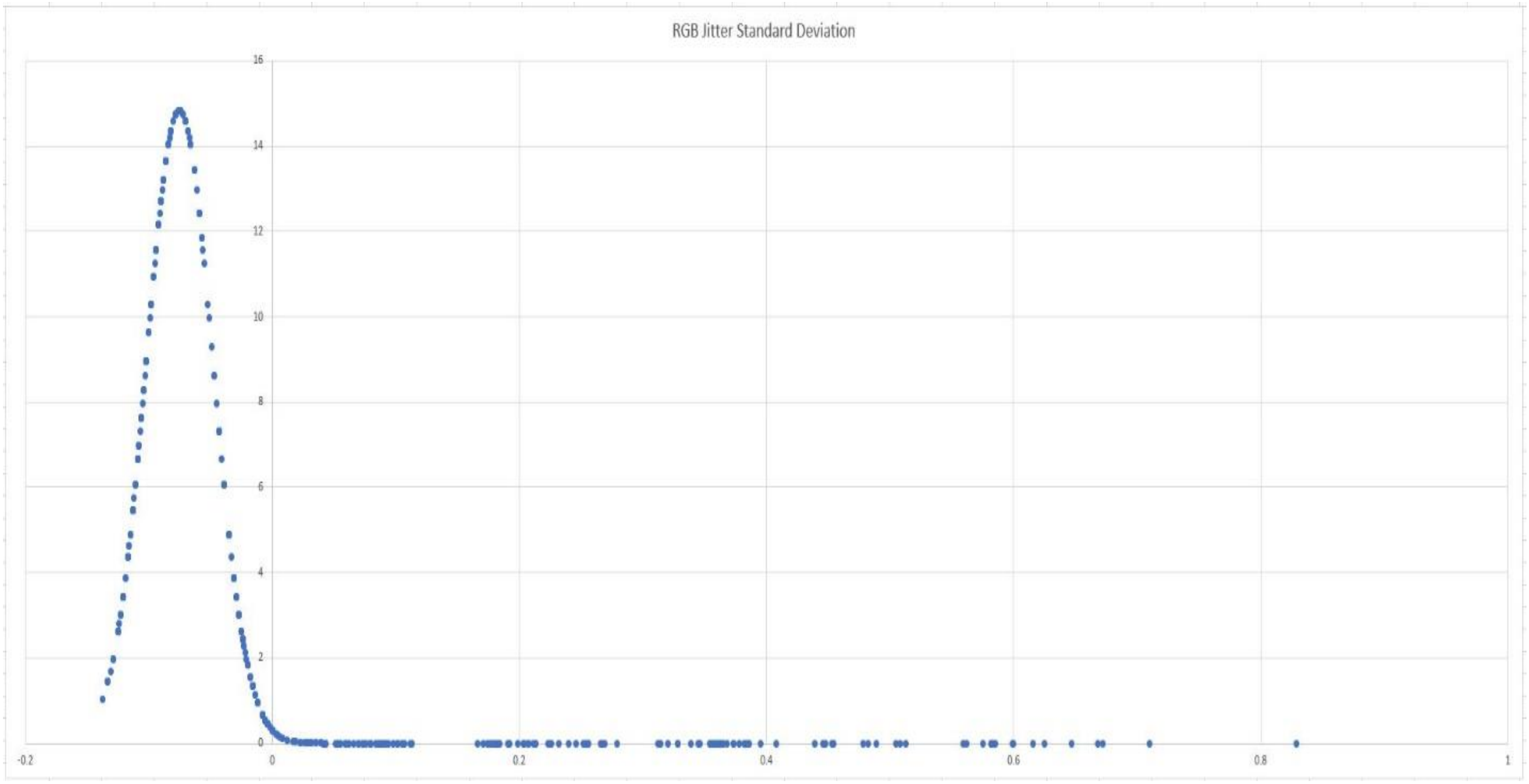
Cam_Monitor Thread Jitter Tracing



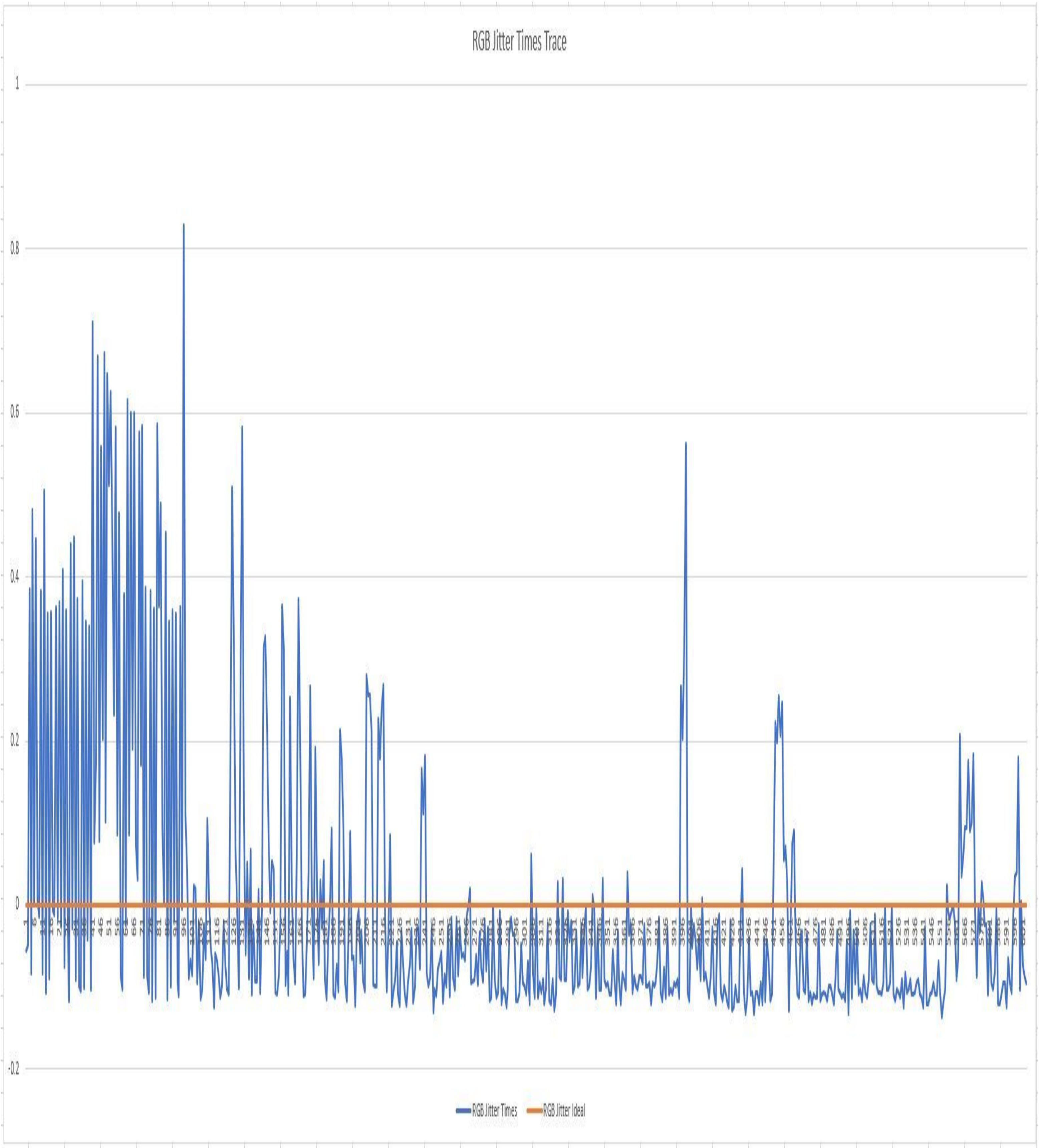
Cam_RGB Thread Jitter Histogram (Part of Profile)



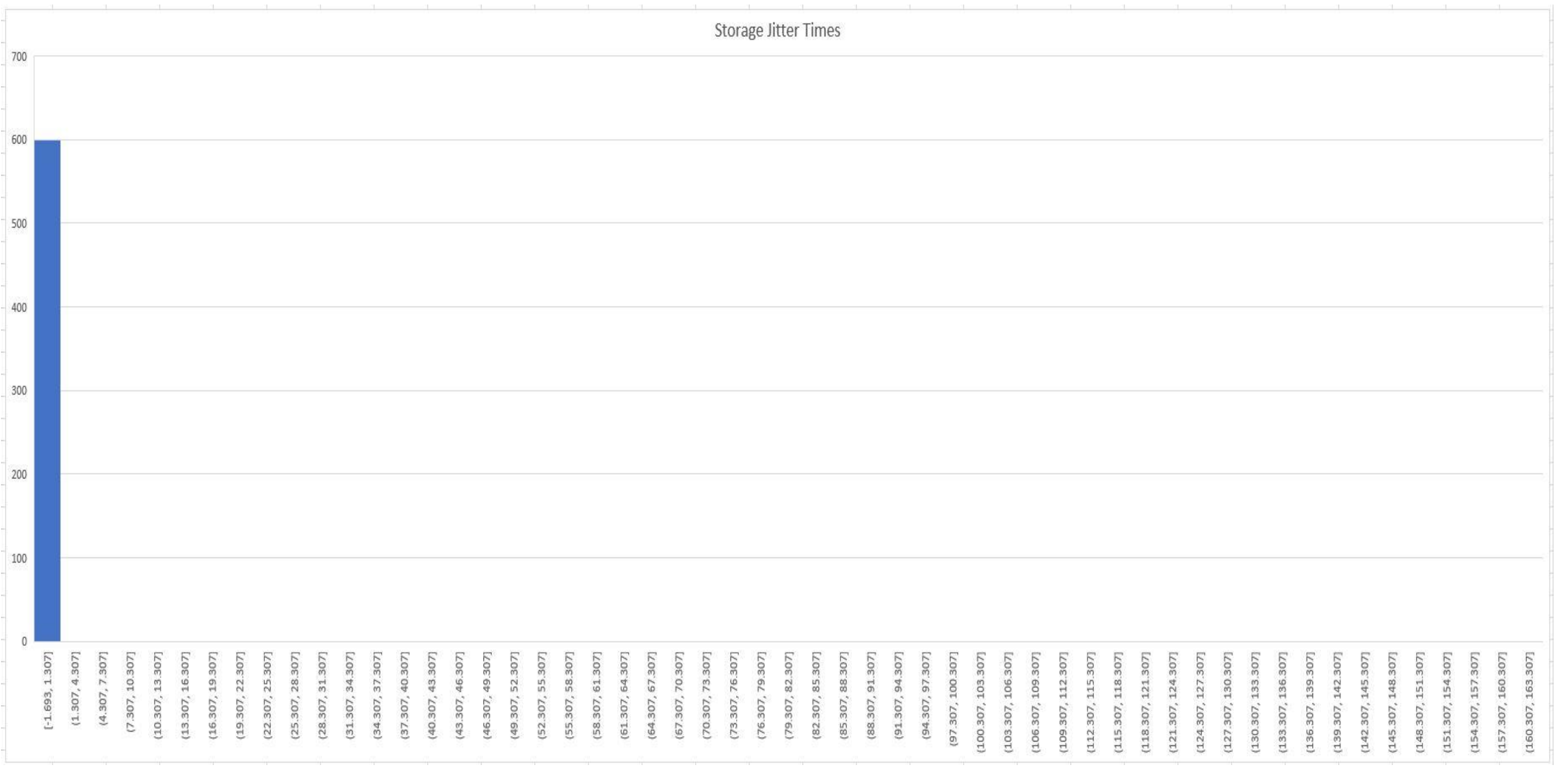
Cam_RGB Thread Jitter Standard Deviation (Part of Profile)



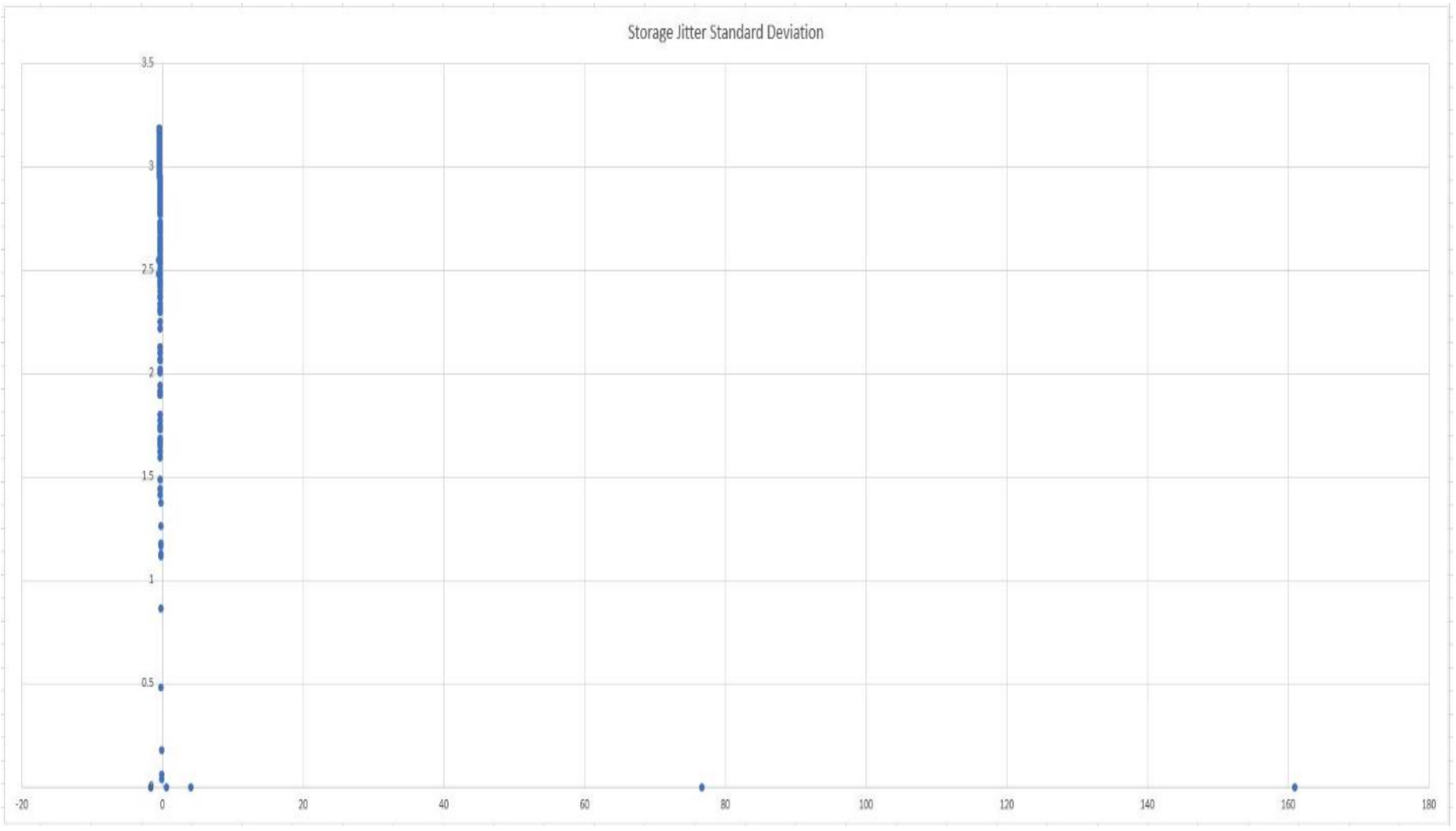
Cam_ RGB Thread Jitter Tracing



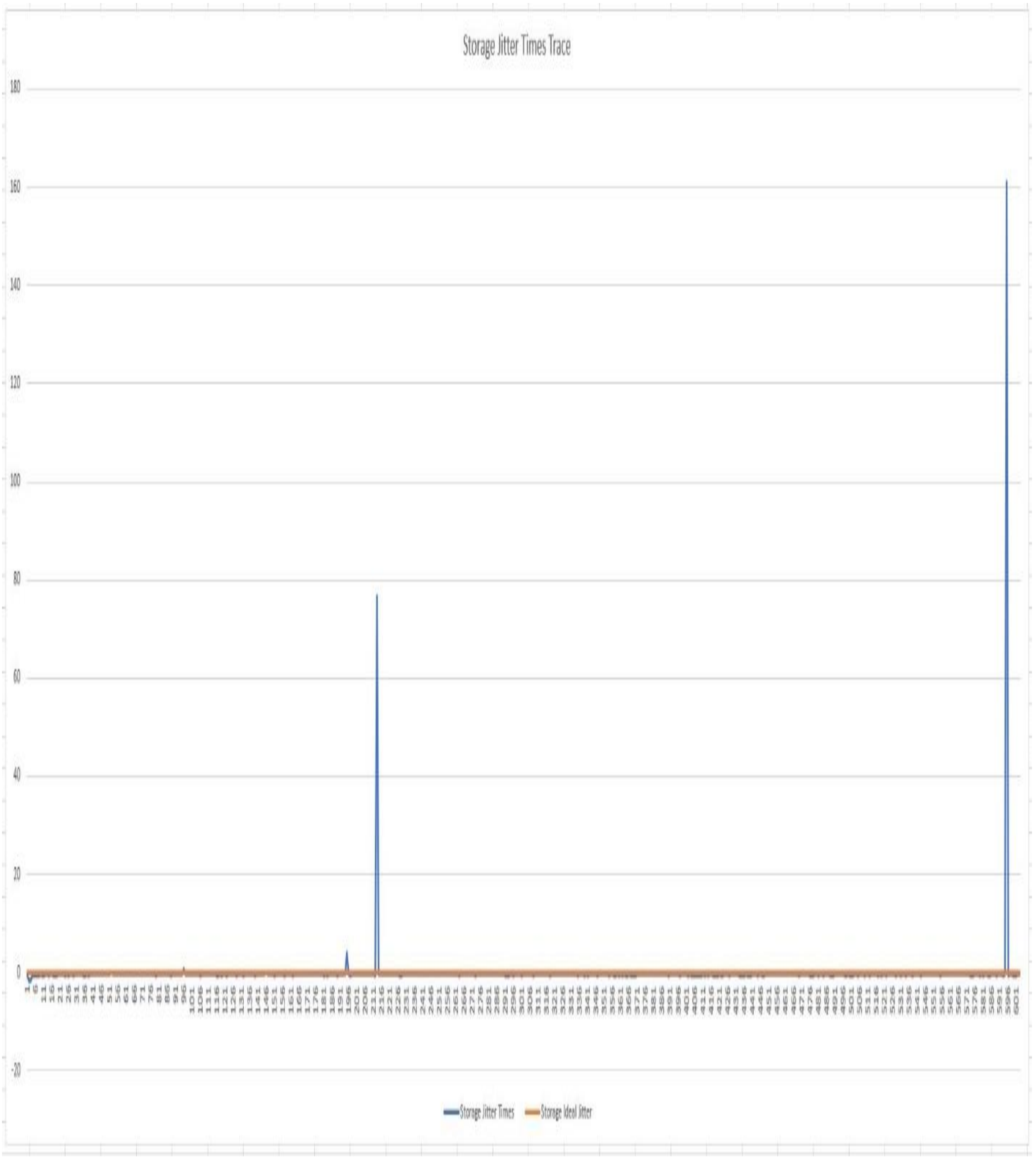
Storage Thread Jitter Histogram (Part of Profile)



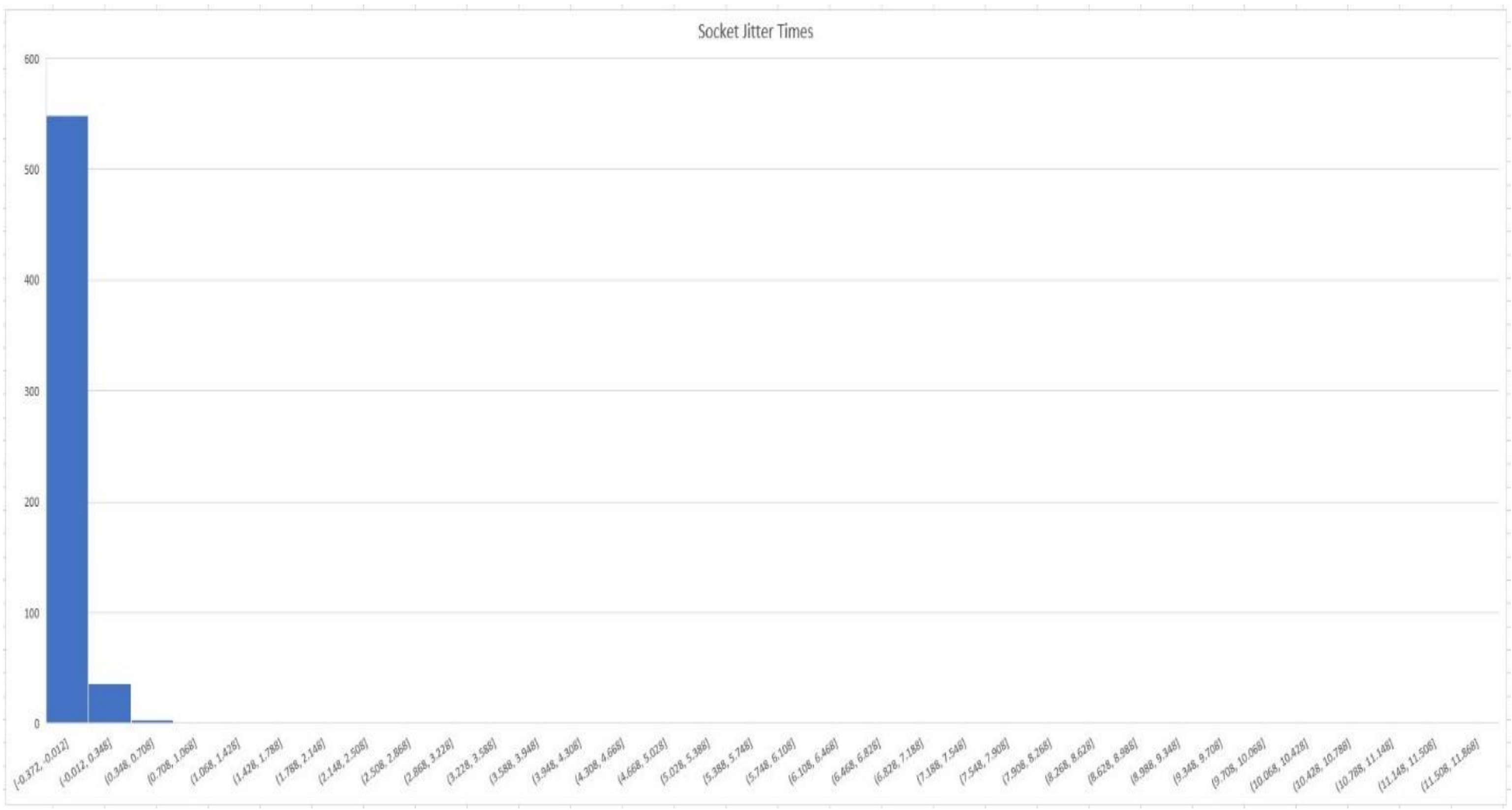
Storage Thread Jitter Standard Deviation (Part of Profile)



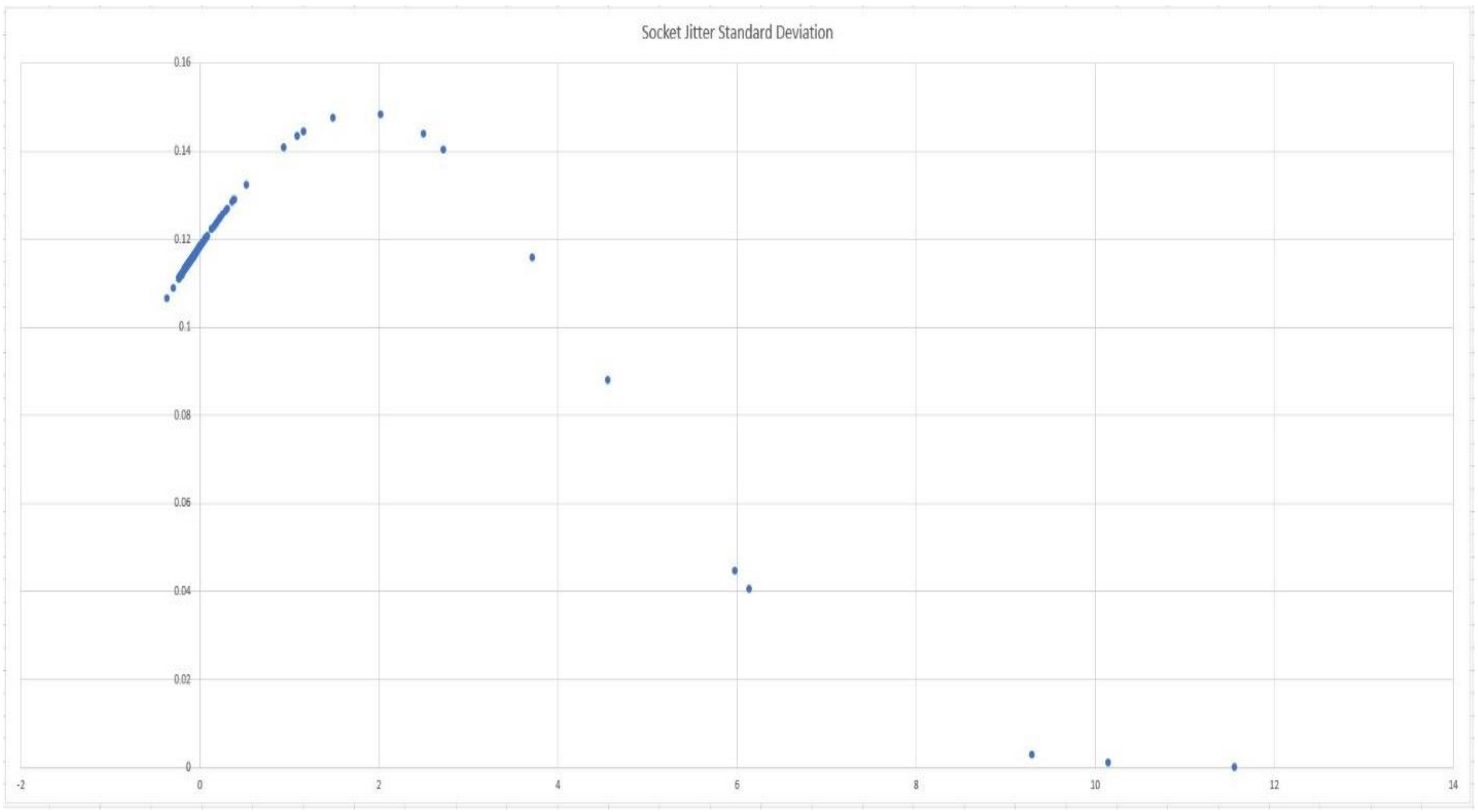
Storage Thread Jitter Tracing



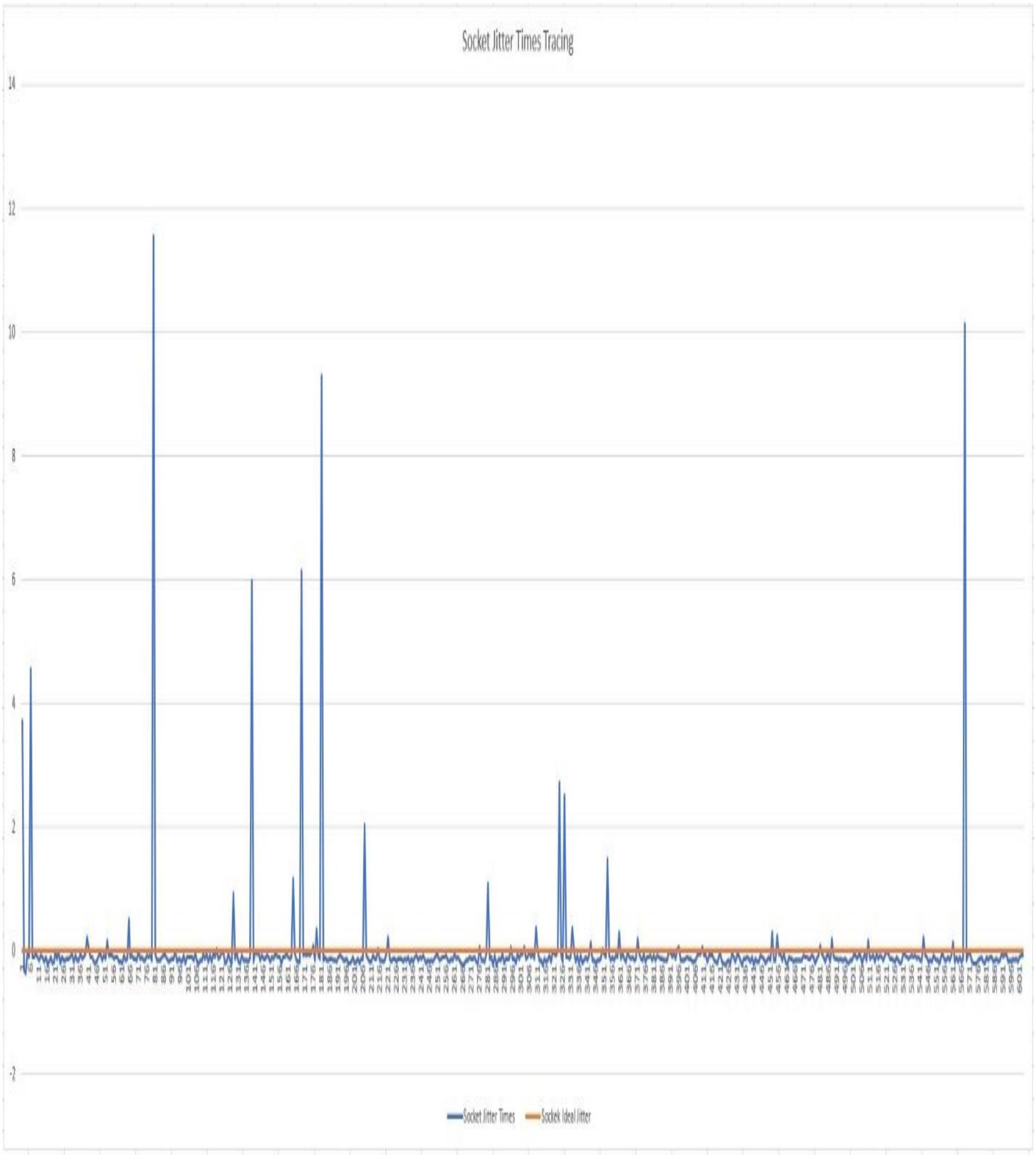
Socket Thread Jitter Histogram (Part of Profile)



Socket Thread Jitter Standard Deviation (Part of Profile)



Socket Thread Jitter Tracing



Monotonic Analysis on 10Hz

<u>This analysis is performed to prove that the digital stopwatch used is faulty, and not the real time system itself. This will be done by comparing theoretical (ideal) timings, to syslog, and the timings being displayed in the stopwatch – on captured Images. All of the information stated below, implements the same.</u>

Without File Transfer

Maximum Positive Δ between Ideal and Syslog: 2.01ms

Minimum Negative Δ between Ideal and Syslog: -0.0059ms

Maximum Positive Δ between Ideal and Stopwatch: 29ms

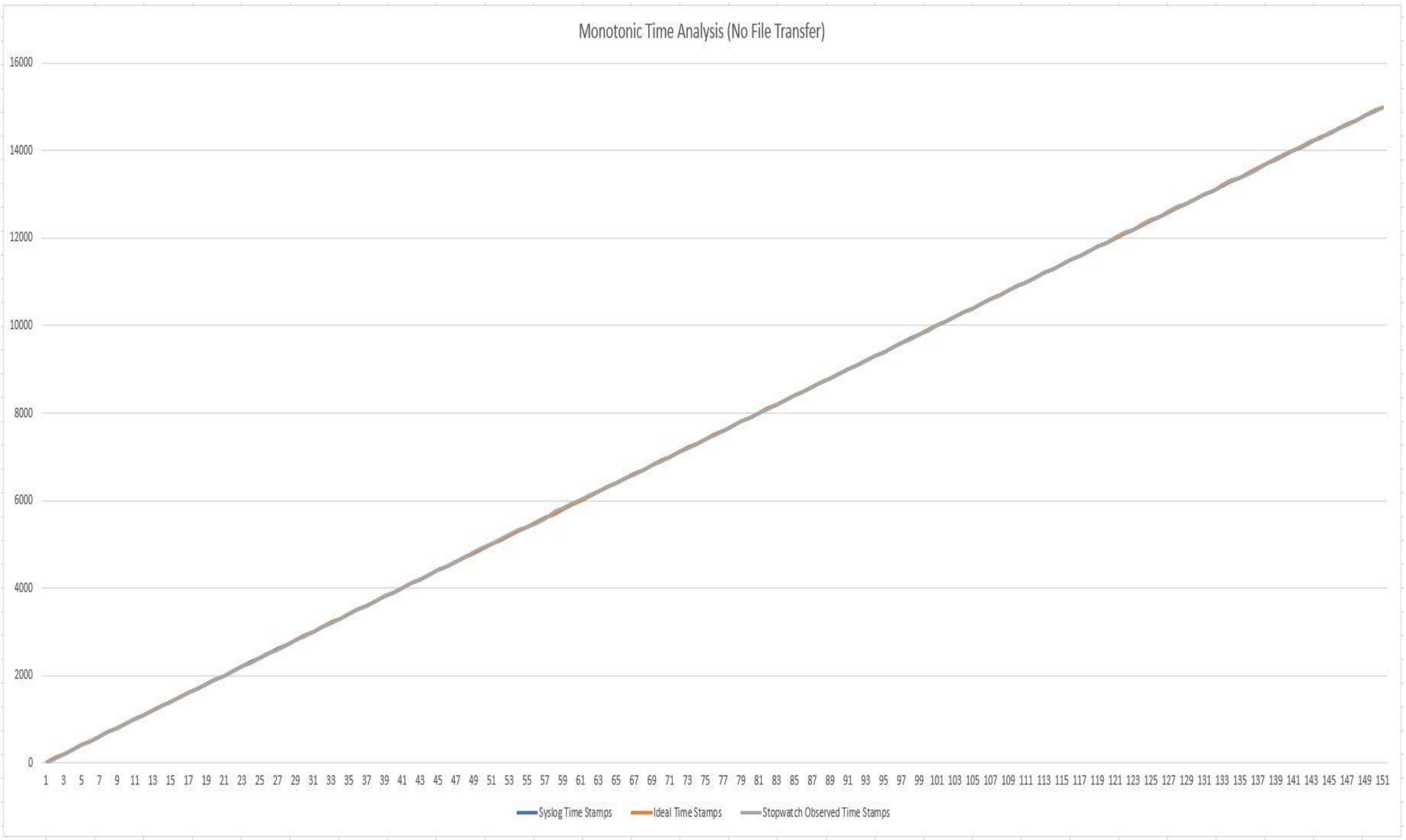
Minimum Negative Δ between Ideal and Stopwatch: -25ms

Maximum Positive Δ between Syslog and Stopwatch: 27ms

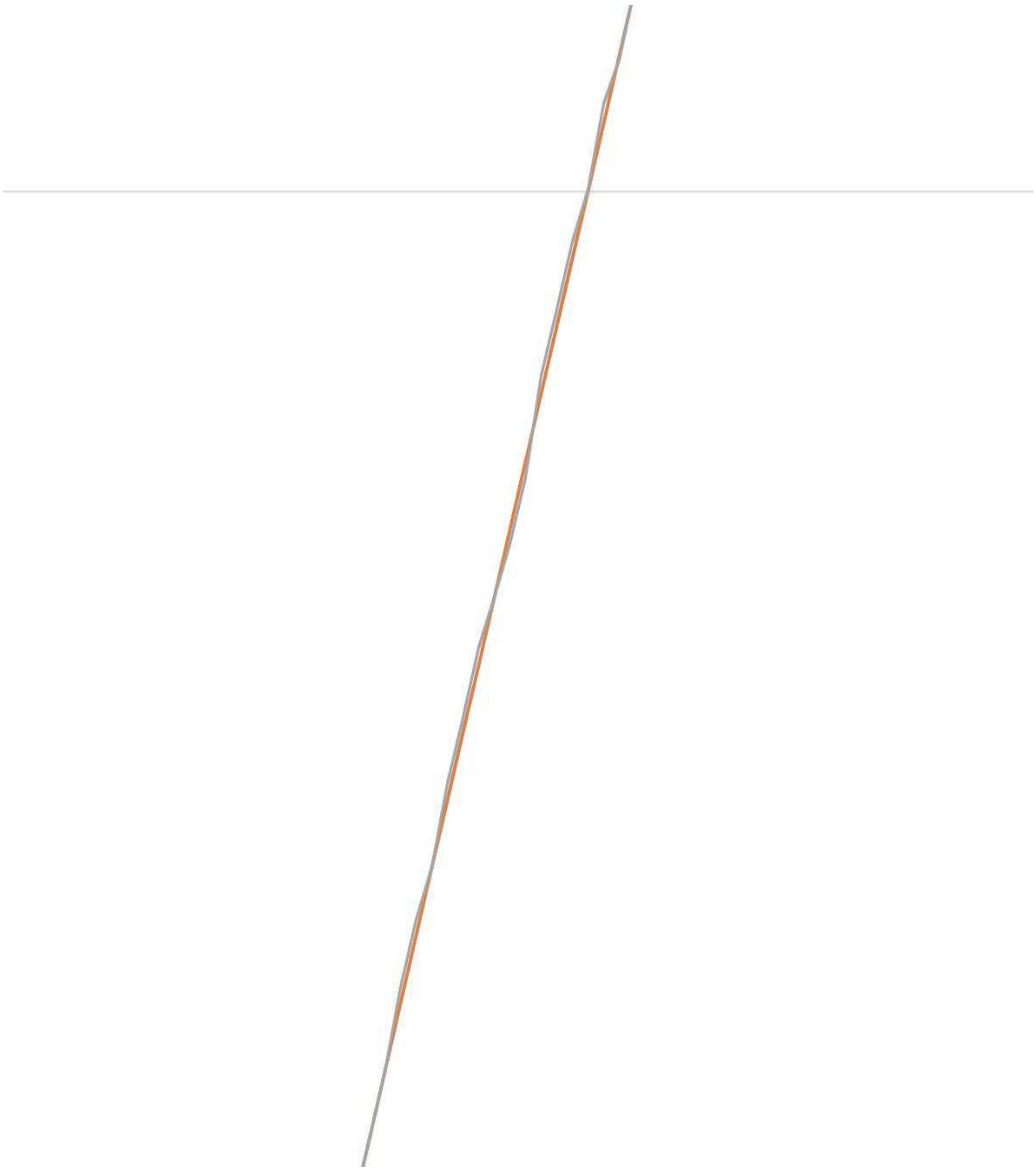
Minimum Negative Δ between Syslog and Stopwatch: -28ms

Evidently, the Syslog stamps are varying only with a millisecond or two, while, the actual clock being used it varying in range more than 50ms – which makes it very hard to take blur free images at 100ms interval

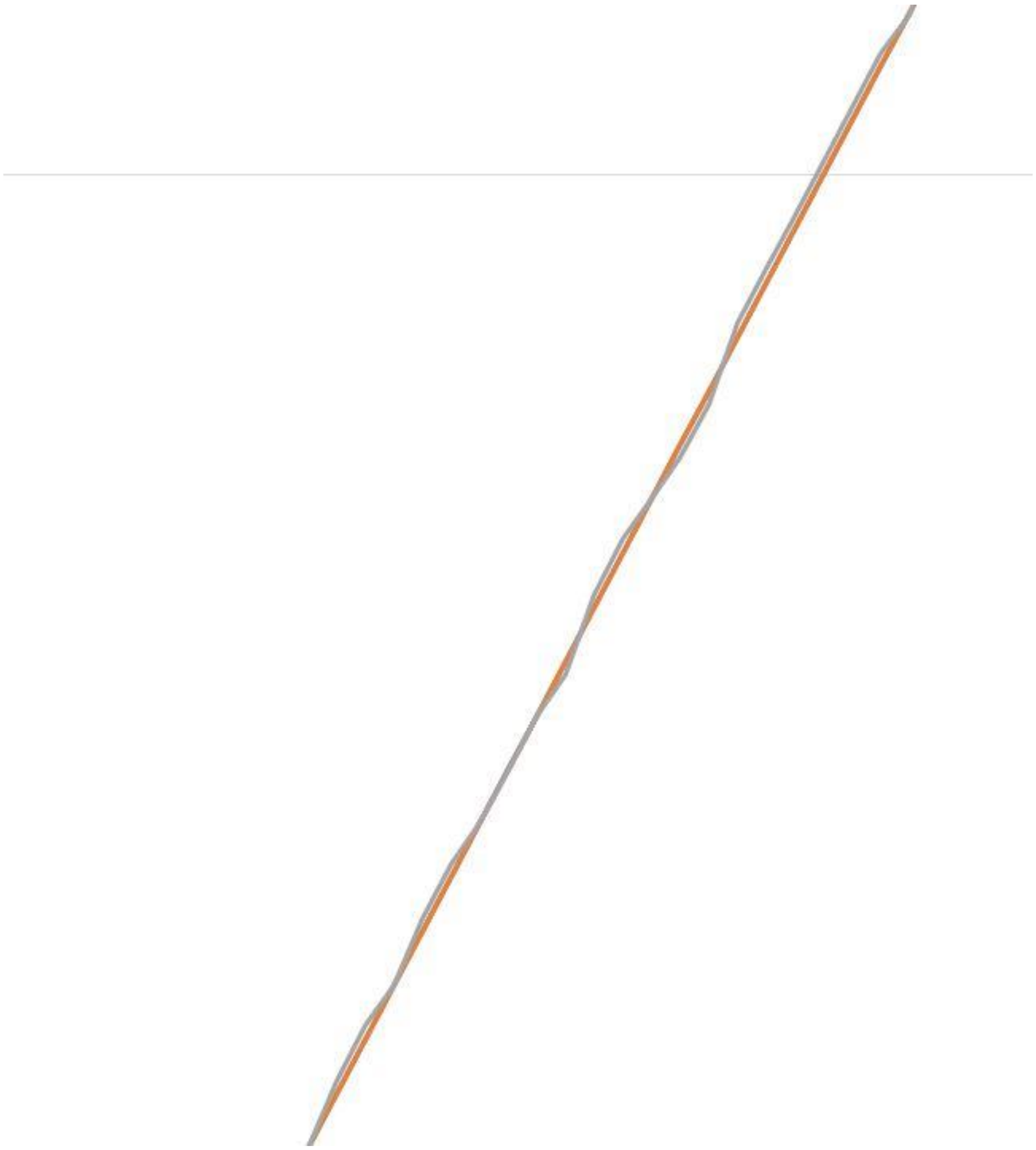
Overall Graph



Region 1 with Observable Errors



Region 2 with Observable Errors



With File Transfer

Maximum Positive Δ between Ideal and Syslog: 0.015ms

Minimum Negative Δ between Ideal and Syslog: -1.048ms

Maximum Positive Δ between Ideal and Stopwatch: 32ms

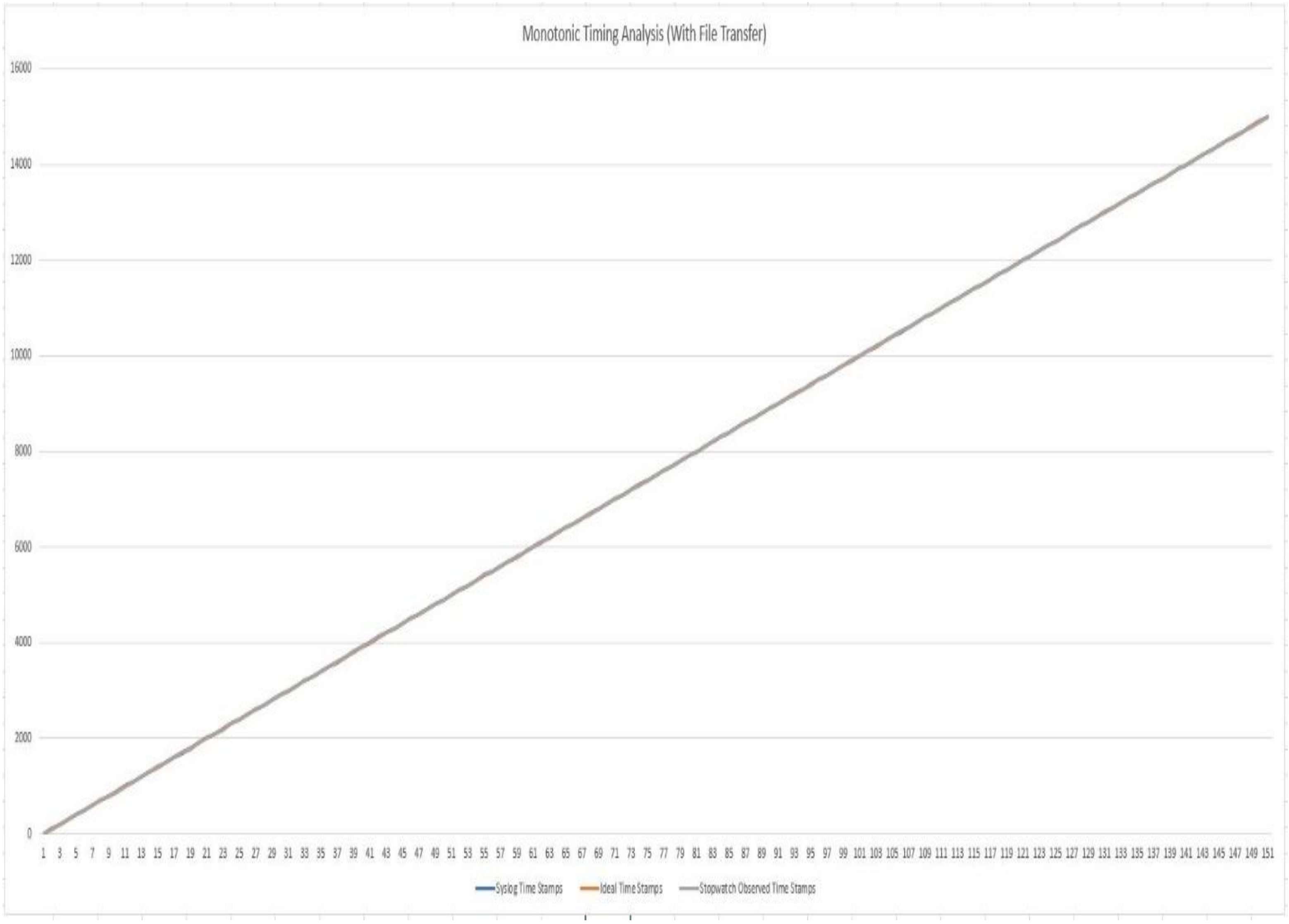
Minimum Negative Δ between Ideal and Stopwatch: -24ms

Maximum Positive Δ between Syslog and Stopwatch: 24ms

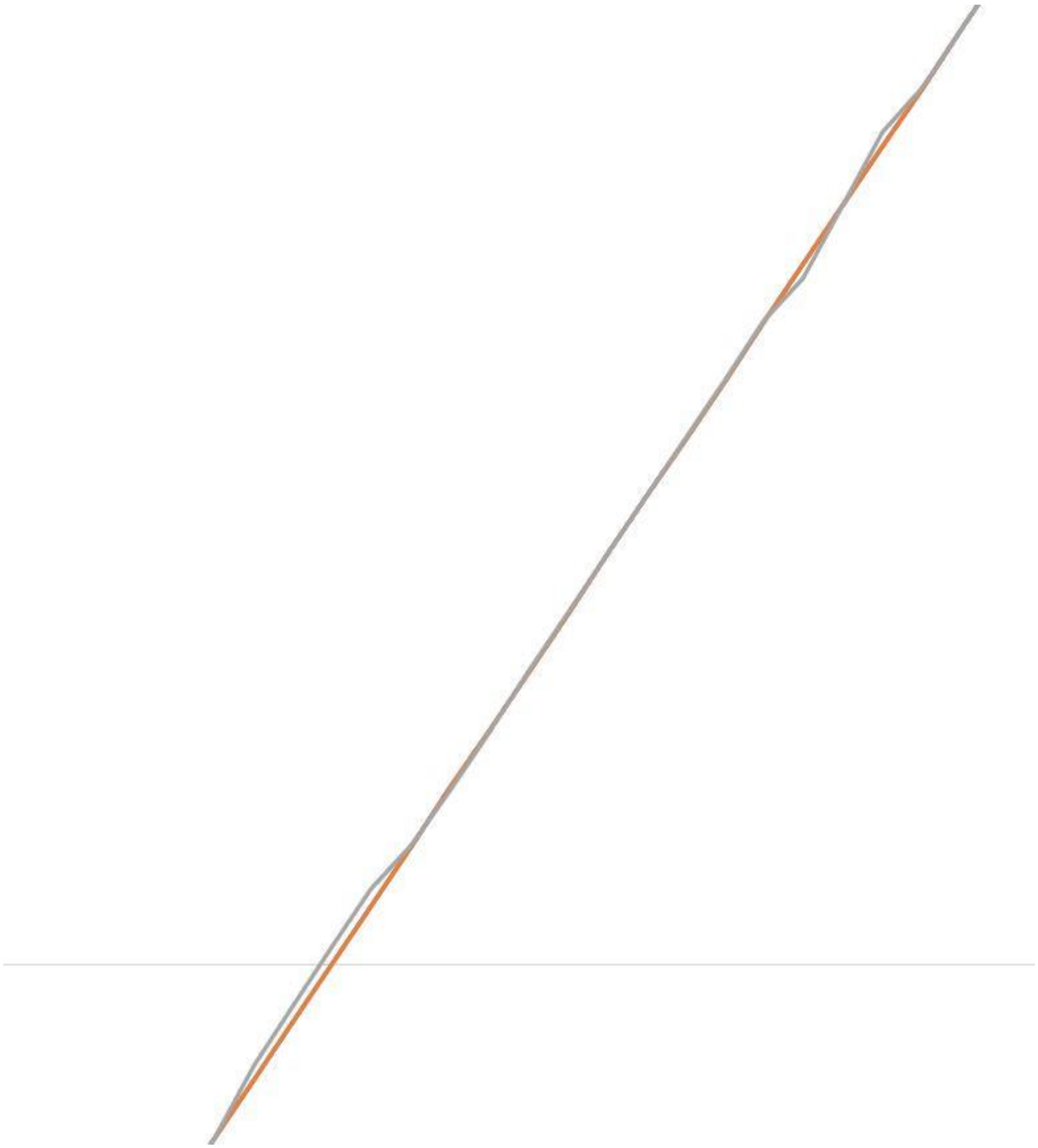
Minimum Negative Δ between Syslog and Stopwatch: -32ms

Evidently, the Syslog stamps are varying only with a millisecond or two, while, the actual clock being used it varying in range more than 50ms – which makes it very hard to take blur free images at 100ms interval. There is no significant change due to file transfer, and that was expected since the CPU cores are different.

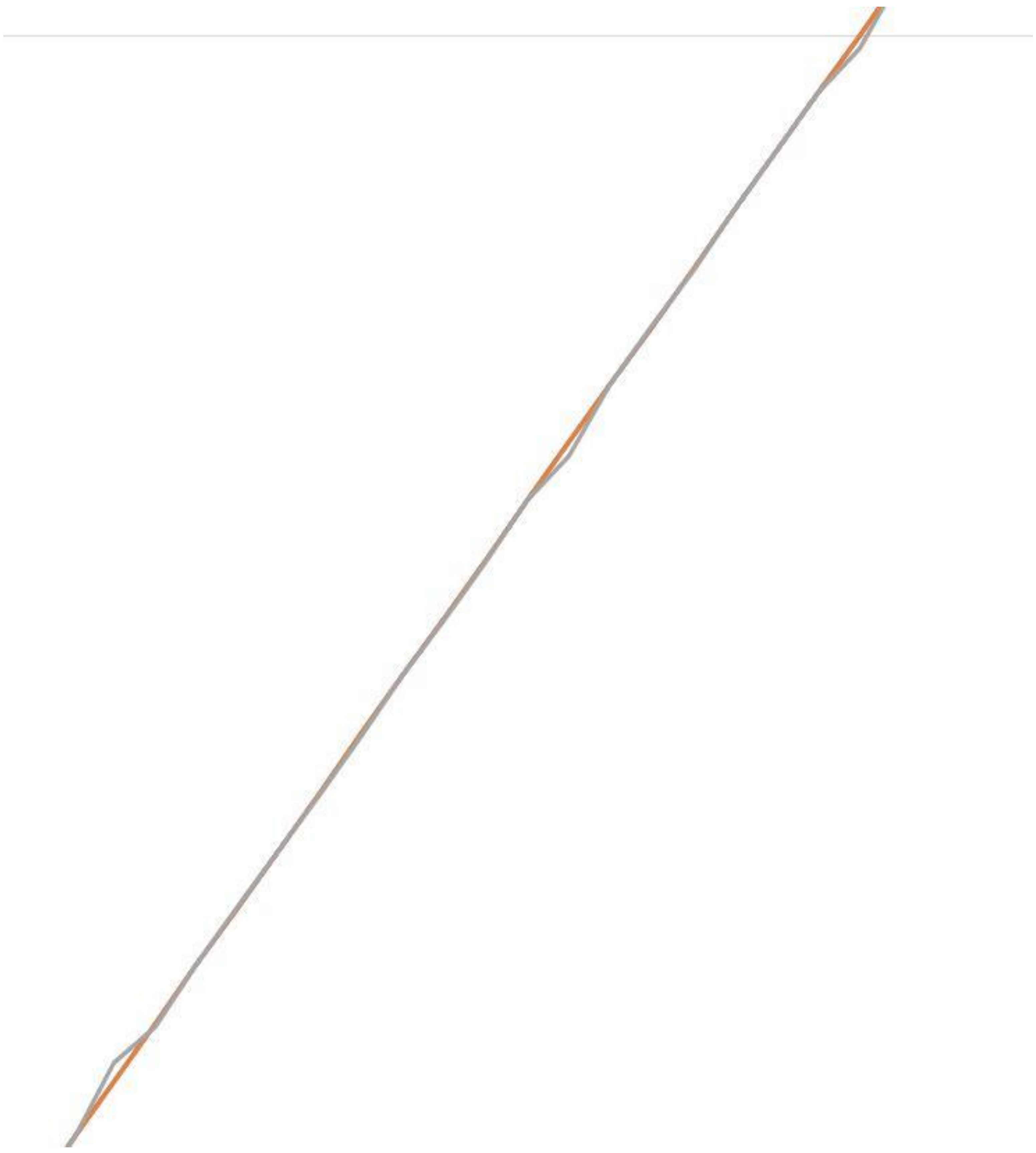
Overall Graph



Region 2 with Observable Errors



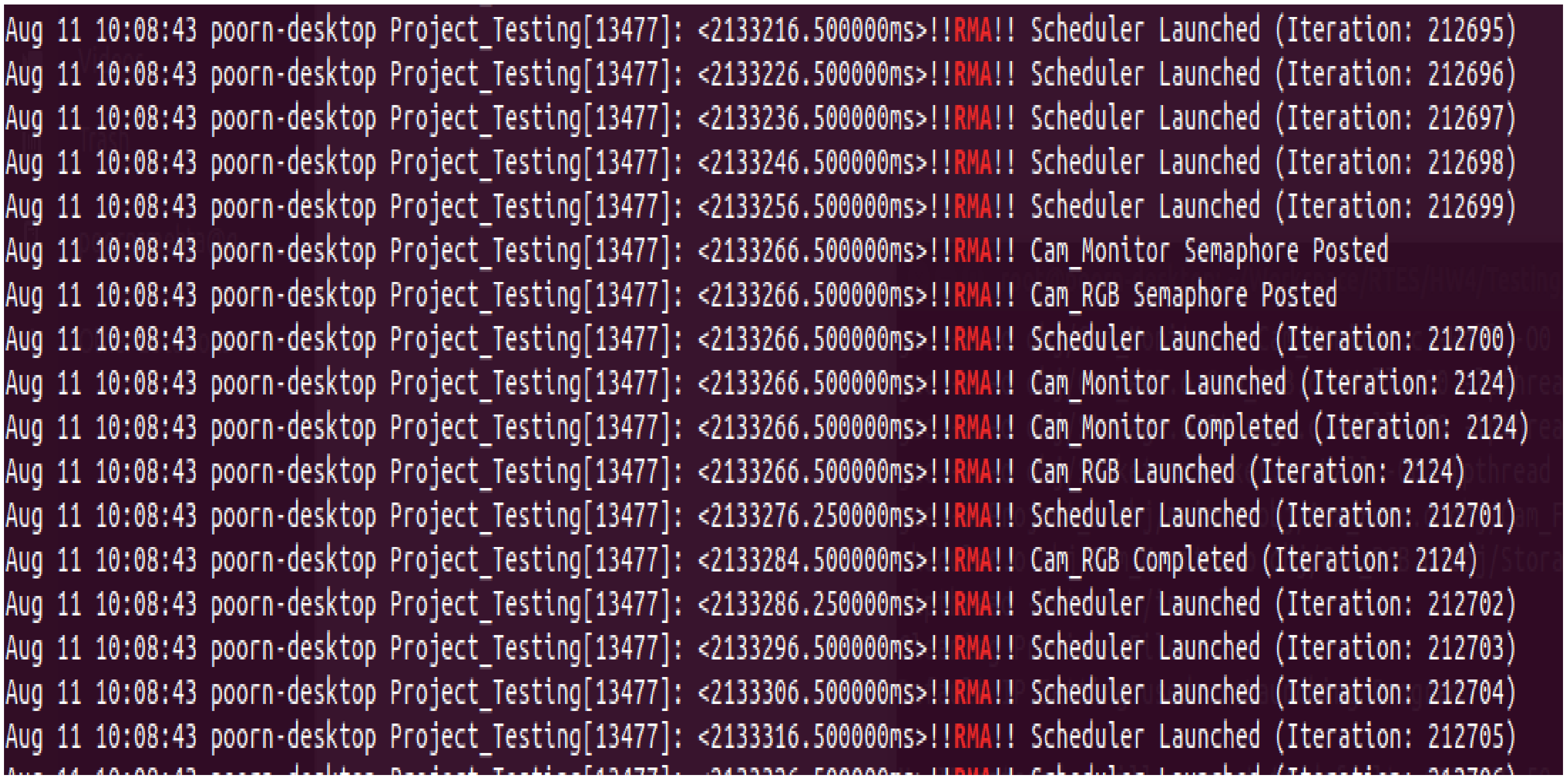
Region 2 with Observable Errors



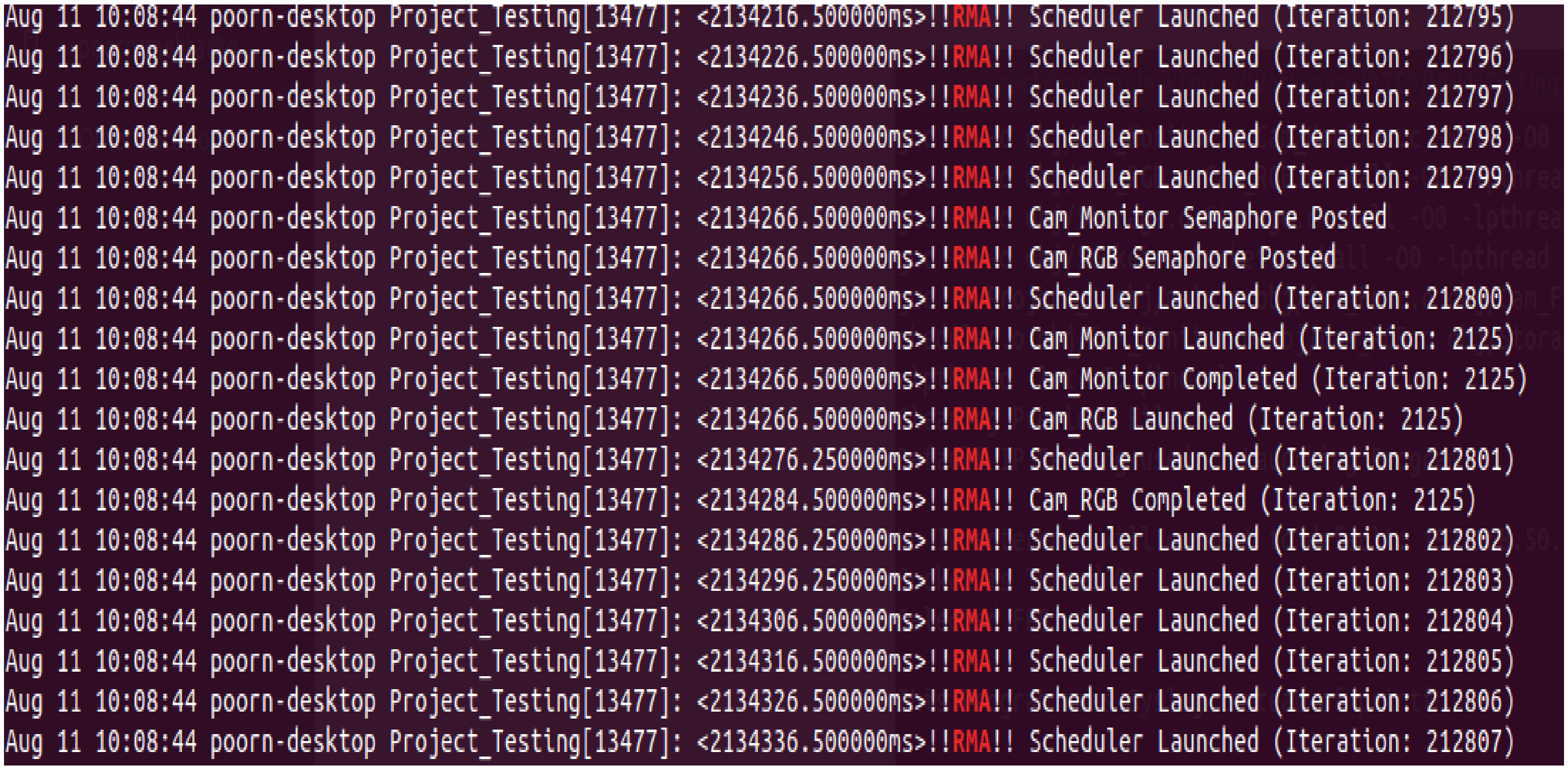
Rate Monotonic Analysis

1Hz Mode

Syslog Output Screenshots

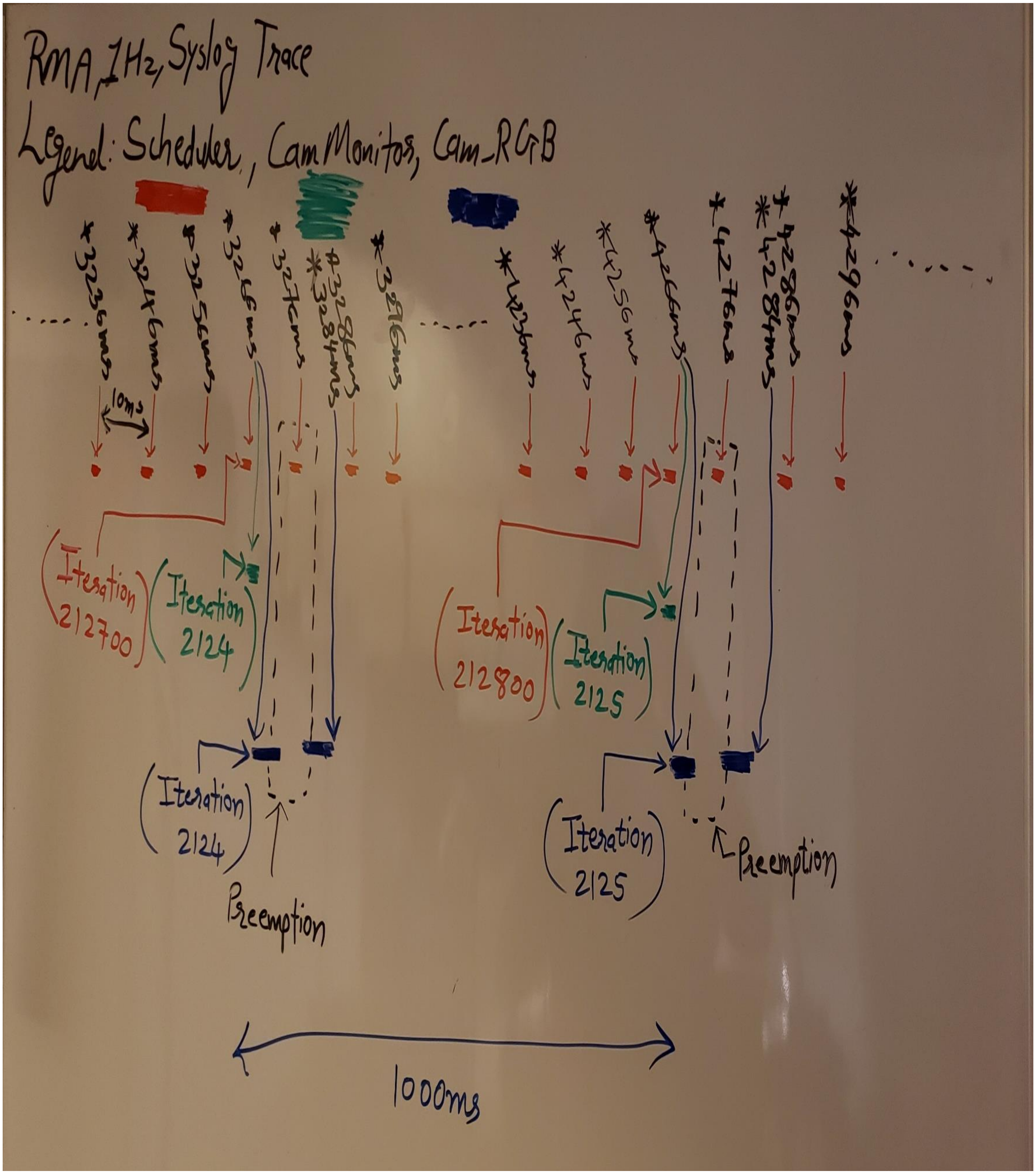


```
Aug 11 10:08:43 poorn-desktop Project_Testing[13477]: <2133216.500000ms>!!RMA!! Scheduler Launched (Iteration: 212695)
Aug 11 10:08:43 poorn-desktop Project_Testing[13477]: <2133226.500000ms>!!RMA!! Scheduler Launched (Iteration: 212696)
Aug 11 10:08:43 poorn-desktop Project_Testing[13477]: <2133236.500000ms>!!RMA!! Scheduler Launched (Iteration: 212697)
Aug 11 10:08:43 poorn-desktop Project_Testing[13477]: <2133246.500000ms>!!RMA!! Scheduler Launched (Iteration: 212698)
Aug 11 10:08:43 poorn-desktop Project_Testing[13477]: <2133256.500000ms>!!RMA!! Scheduler Launched (Iteration: 212699)
Aug 11 10:08:43 poorn-desktop Project_Testing[13477]: <2133266.500000ms>!!RMA!! Cam_Monitor Semaphore Posted
Aug 11 10:08:43 poorn-desktop Project_Testing[13477]: <2133266.500000ms>!!RMA!! Cam_RGB Semaphore Posted
Aug 11 10:08:43 poorn-desktop Project_Testing[13477]: <2133266.500000ms>!!RMA!! Scheduler Launched (Iteration: 212700)
Aug 11 10:08:43 poorn-desktop Project_Testing[13477]: <2133266.500000ms>!!RMA!! Cam_Monitor Launched (Iteration: 2124)
Aug 11 10:08:43 poorn-desktop Project_Testing[13477]: <2133266.500000ms>!!RMA!! Cam_Monitor Completed (Iteration: 2124)
Aug 11 10:08:43 poorn-desktop Project_Testing[13477]: <2133266.500000ms>!!RMA!! Cam_RGB Launched (Iteration: 2124)
Aug 11 10:08:43 poorn-desktop Project_Testing[13477]: <2133276.250000ms>!!RMA!! Scheduler Launched (Iteration: 212701)
Aug 11 10:08:43 poorn-desktop Project_Testing[13477]: <2133284.500000ms>!!RMA!! Cam_RGB Completed (Iteration: 2124)
Aug 11 10:08:43 poorn-desktop Project_Testing[13477]: <2133286.250000ms>!!RMA!! Scheduler Launched (Iteration: 212702)
Aug 11 10:08:43 poorn-desktop Project_Testing[13477]: <2133296.500000ms>!!RMA!! Scheduler Launched (Iteration: 212703)
Aug 11 10:08:43 poorn-desktop Project_Testing[13477]: <2133306.500000ms>!!RMA!! Scheduler Launched (Iteration: 212704)
Aug 11 10:08:43 poorn-desktop Project_Testing[13477]: <2133316.500000ms>!!RMA!! Scheduler Launched (Iteration: 212705)
```



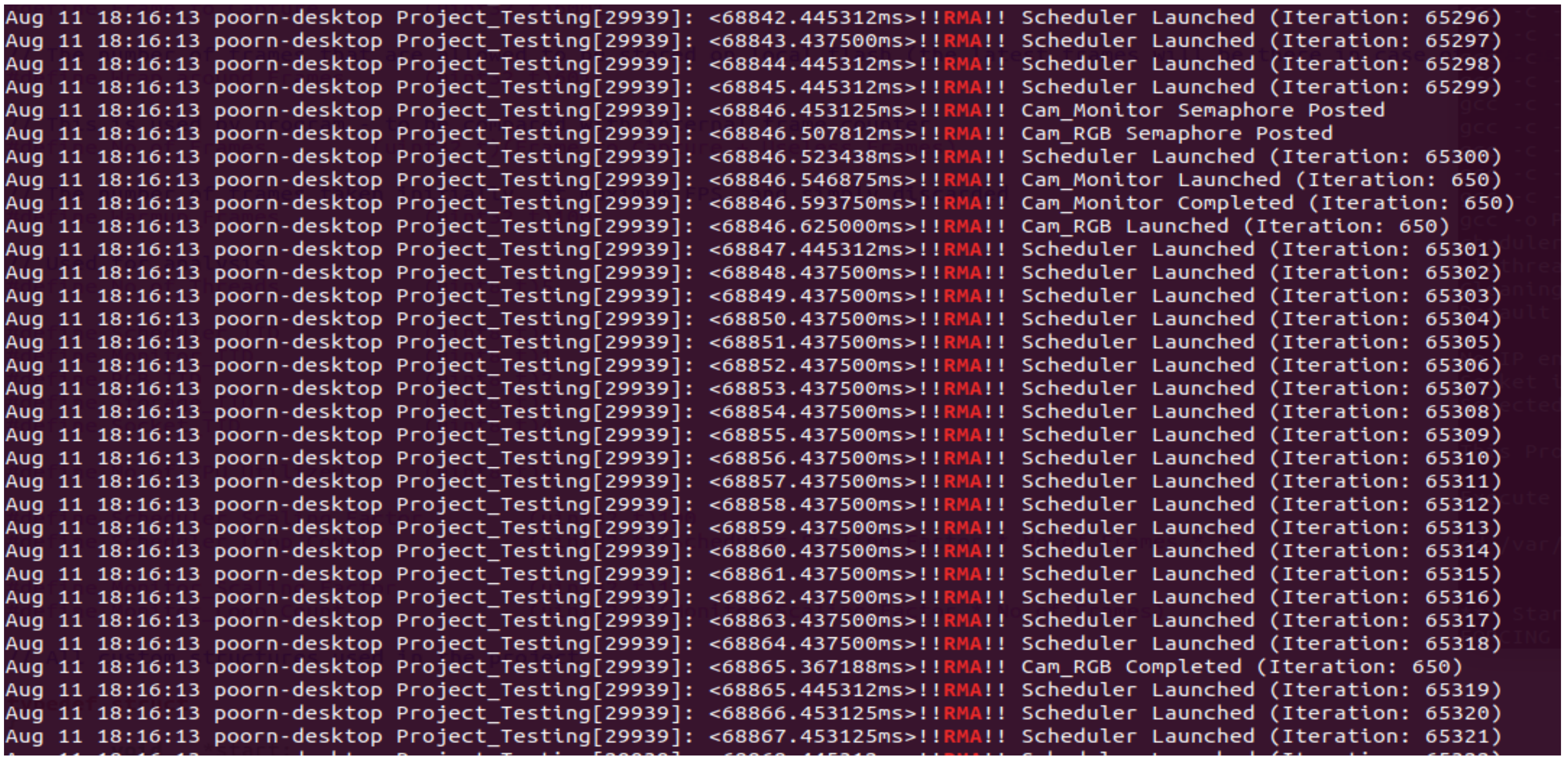
```
Aug 11 10:08:44 poorn-desktop Project_Testing[13477]: <2134216.500000ms>!!RMA!! Scheduler Launched (Iteration: 212795)
Aug 11 10:08:44 poorn-desktop Project_Testing[13477]: <2134226.500000ms>!!RMA!! Scheduler Launched (Iteration: 212796)
Aug 11 10:08:44 poorn-desktop Project_Testing[13477]: <2134236.500000ms>!!RMA!! Scheduler Launched (Iteration: 212797)
Aug 11 10:08:44 poorn-desktop Project_Testing[13477]: <2134246.500000ms>!!RMA!! Scheduler Launched (Iteration: 212798)
Aug 11 10:08:44 poorn-desktop Project_Testing[13477]: <2134256.500000ms>!!RMA!! Scheduler Launched (Iteration: 212799)
Aug 11 10:08:44 poorn-desktop Project_Testing[13477]: <2134266.500000ms>!!RMA!! Cam_Monitor Semaphore Posted
Aug 11 10:08:44 poorn-desktop Project_Testing[13477]: <2134266.500000ms>!!RMA!! Cam_RGB Semaphore Posted
Aug 11 10:08:44 poorn-desktop Project_Testing[13477]: <2134266.500000ms>!!RMA!! Scheduler Launched (Iteration: 212800)
Aug 11 10:08:44 poorn-desktop Project_Testing[13477]: <2134266.500000ms>!!RMA!! Cam_Monitor Launched (Iteration: 2125)
Aug 11 10:08:44 poorn-desktop Project_Testing[13477]: <2134266.500000ms>!!RMA!! Cam_Monitor Completed (Iteration: 2125)
Aug 11 10:08:44 poorn-desktop Project_Testing[13477]: <2134266.500000ms>!!RMA!! Cam_RGB Launched (Iteration: 2125)
Aug 11 10:08:44 poorn-desktop Project_Testing[13477]: <2134276.250000ms>!!RMA!! Scheduler Launched (Iteration: 212801)
Aug 11 10:08:44 poorn-desktop Project_Testing[13477]: <2134284.500000ms>!!RMA!! Cam_RGB Completed (Iteration: 2125)
Aug 11 10:08:44 poorn-desktop Project_Testing[13477]: <2134286.250000ms>!!RMA!! Scheduler Launched (Iteration: 212802)
Aug 11 10:08:44 poorn-desktop Project_Testing[13477]: <2134296.250000ms>!!RMA!! Scheduler Launched (Iteration: 212803)
Aug 11 10:08:44 poorn-desktop Project_Testing[13477]: <2134306.500000ms>!!RMA!! Scheduler Launched (Iteration: 212804)
Aug 11 10:08:44 poorn-desktop Project_Testing[13477]: <2134316.500000ms>!!RMA!! Scheduler Launched (Iteration: 212805)
Aug 11 10:08:44 poorn-desktop Project_Testing[13477]: <2134326.500000ms>!!RMA!! Scheduler Launched (Iteration: 212806)
Aug 11 10:08:44 poorn-desktop Project_Testing[13477]: <2134336.500000ms>!!RMA!! Scheduler Launched (Iteration: 212807)
```

RMA Based on Syslog

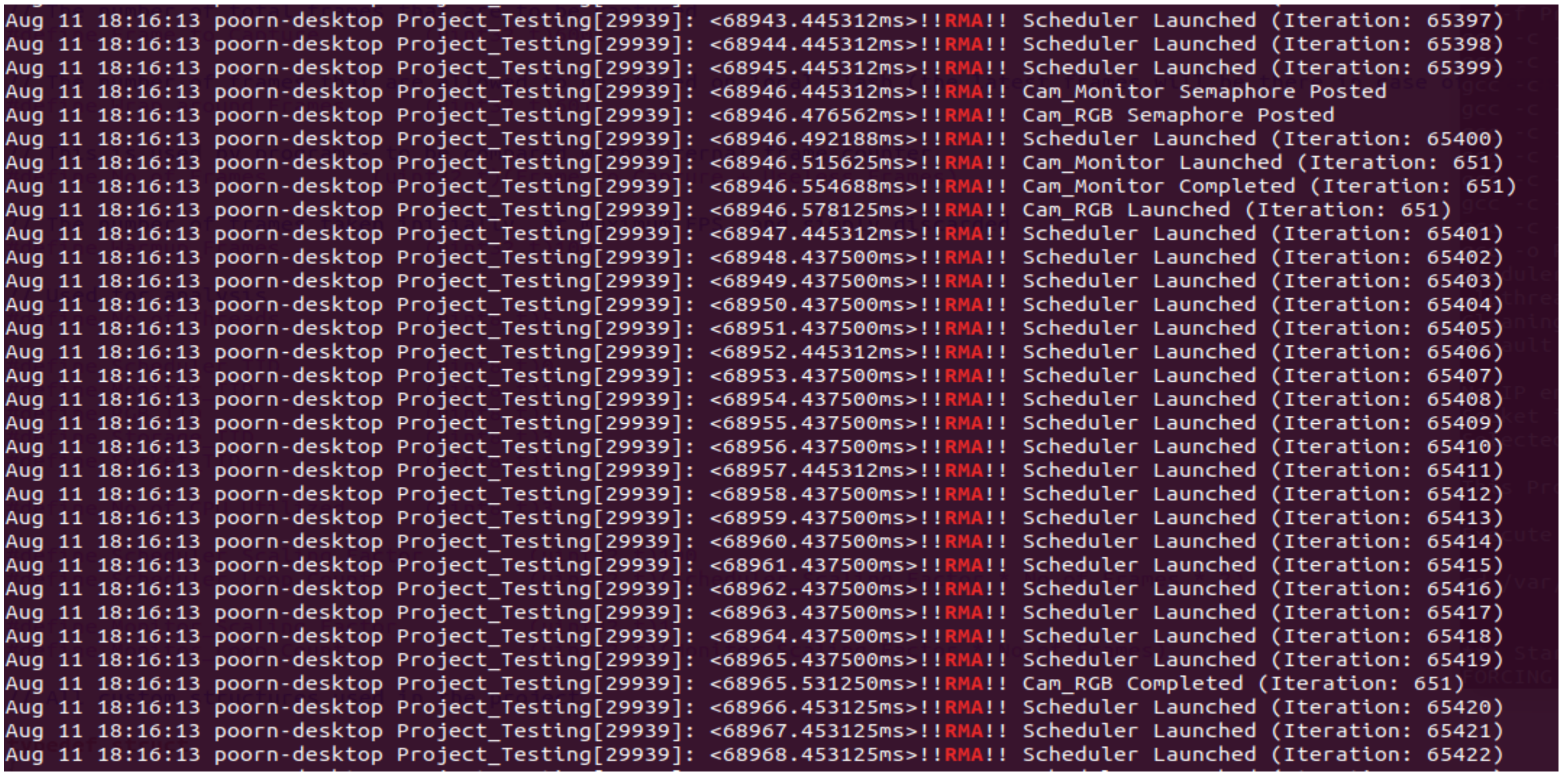


10Hz Mode

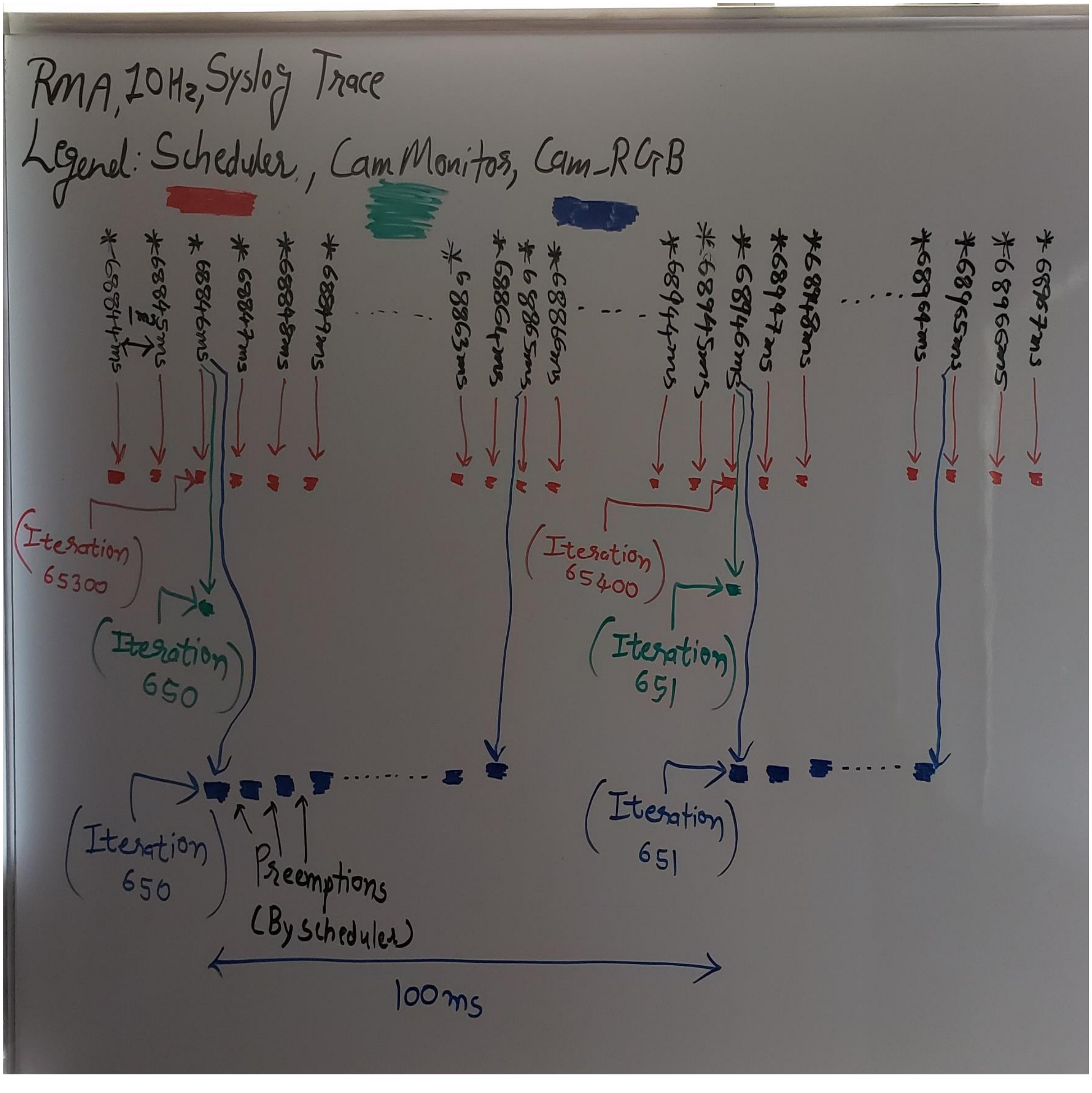
Syslog Output Screenshots



```
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68842.445312ms>!!RMA!! Scheduler Launched (Iteration: 65296)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68843.437500ms>!!RMA!! Scheduler Launched (Iteration: 65297)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68844.445312ms>!!RMA!! Scheduler Launched (Iteration: 65298)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68845.445312ms>!!RMA!! Scheduler Launched (Iteration: 65299)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68846.453125ms>!!RMA!! Cam_Monitor Semaphore Posted
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68846.507812ms>!!RMA!! Cam_RGB Semaphore Posted
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68846.523438ms>!!RMA!! Scheduler Launched (Iteration: 65300)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68846.546875ms>!!RMA!! Cam_Monitor Launched (Iteration: 650)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68846.593750ms>!!RMA!! Cam_Monitor Completed (Iteration: 650)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68846.625000ms>!!RMA!! Cam_RGB Launched (Iteration: 650)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68847.445312ms>!!RMA!! Scheduler Launched (Iteration: 65301)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68848.437500ms>!!RMA!! Scheduler Launched (Iteration: 65302)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68849.437500ms>!!RMA!! Scheduler Launched (Iteration: 65303)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68850.437500ms>!!RMA!! Scheduler Launched (Iteration: 65304)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68851.437500ms>!!RMA!! Scheduler Launched (Iteration: 65305)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68852.437500ms>!!RMA!! Scheduler Launched (Iteration: 65306)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68853.437500ms>!!RMA!! Scheduler Launched (Iteration: 65307)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68854.437500ms>!!RMA!! Scheduler Launched (Iteration: 65308)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68855.437500ms>!!RMA!! Scheduler Launched (Iteration: 65309)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68856.437500ms>!!RMA!! Scheduler Launched (Iteration: 65310)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68857.437500ms>!!RMA!! Scheduler Launched (Iteration: 65311)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68858.437500ms>!!RMA!! Scheduler Launched (Iteration: 65312)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68859.437500ms>!!RMA!! Scheduler Launched (Iteration: 65313)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68860.437500ms>!!RMA!! Scheduler Launched (Iteration: 65314)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68861.437500ms>!!RMA!! Scheduler Launched (Iteration: 65315)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68862.437500ms>!!RMA!! Scheduler Launched (Iteration: 65316)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68863.437500ms>!!RMA!! Scheduler Launched (Iteration: 65317)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68864.437500ms>!!RMA!! Scheduler Launched (Iteration: 65318)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68865.367188ms>!!RMA!! Cam_RGB Completed (Iteration: 650)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68865.445312ms>!!RMA!! Scheduler Launched (Iteration: 65319)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68866.453125ms>!!RMA!! Scheduler Launched (Iteration: 65320)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68867.453125ms>!!RMA!! Scheduler Launched (Iteration: 65321)
```

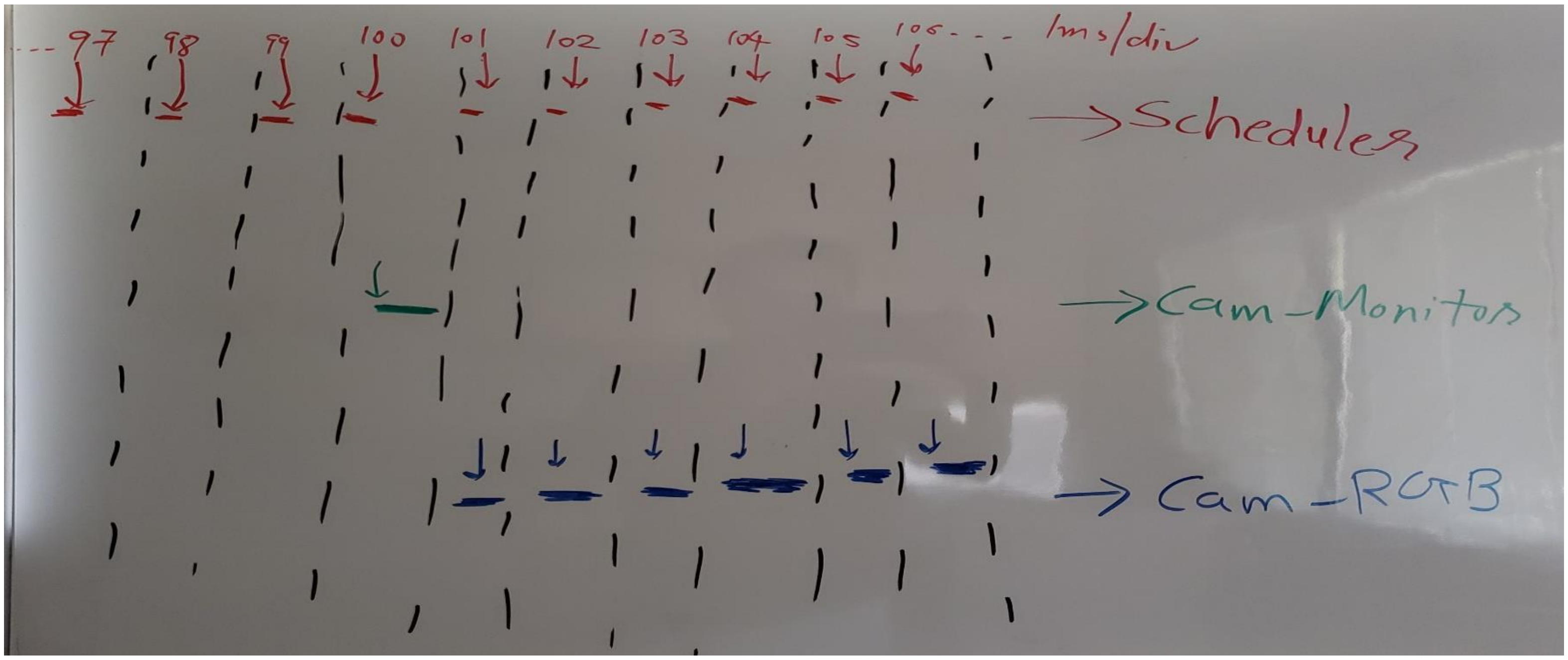


```
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68943.445312ms>!!RMA!! Scheduler Launched (Iteration: 65397)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68944.445312ms>!!RMA!! Scheduler Launched (Iteration: 65398)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68945.445312ms>!!RMA!! Scheduler Launched (Iteration: 65399)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68946.445312ms>!!RMA!! Cam_Monitor Semaphore Posted
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68946.476562ms>!!RMA!! Cam_RGB Semaphore Posted
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68946.492188ms>!!RMA!! Scheduler Launched (Iteration: 65400)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68946.515625ms>!!RMA!! Cam_Monitor Launched (Iteration: 651)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68946.554688ms>!!RMA!! Cam_Monitor Completed (Iteration: 651)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68946.578125ms>!!RMA!! Cam_RGB Launched (Iteration: 651)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68947.445312ms>!!RMA!! Scheduler Launched (Iteration: 65401)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68948.437500ms>!!RMA!! Scheduler Launched (Iteration: 65402)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68949.437500ms>!!RMA!! Scheduler Launched (Iteration: 65403)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68950.437500ms>!!RMA!! Scheduler Launched (Iteration: 65404)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68951.437500ms>!!RMA!! Scheduler Launched (Iteration: 65405)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68952.445312ms>!!RMA!! Scheduler Launched (Iteration: 65406)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68953.437500ms>!!RMA!! Scheduler Launched (Iteration: 65407)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68954.437500ms>!!RMA!! Scheduler Launched (Iteration: 65408)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68955.437500ms>!!RMA!! Scheduler Launched (Iteration: 65409)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68956.437500ms>!!RMA!! Scheduler Launched (Iteration: 65410)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68957.445312ms>!!RMA!! Scheduler Launched (Iteration: 65411)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68958.437500ms>!!RMA!! Scheduler Launched (Iteration: 65412)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68959.437500ms>!!RMA!! Scheduler Launched (Iteration: 65413)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68960.437500ms>!!RMA!! Scheduler Launched (Iteration: 65414)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68961.437500ms>!!RMA!! Scheduler Launched (Iteration: 65415)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68962.437500ms>!!RMA!! Scheduler Launched (Iteration: 65416)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68963.437500ms>!!RMA!! Scheduler Launched (Iteration: 65417)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68964.437500ms>!!RMA!! Scheduler Launched (Iteration: 65418)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68965.437500ms>!!RMA!! Scheduler Launched (Iteration: 65419)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68965.531250ms>!!RMA!! Cam_RGB Completed (Iteration: 651)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68966.453125ms>!!RMA!! Scheduler Launched (Iteration: 65420)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68967.453125ms>!!RMA!! Scheduler Launched (Iteration: 65421)
Aug 11 18:16:13 poorn-desktop Project_Testing[29939]: <68968.453125ms>!!RMA!! Scheduler Launched (Iteration: 65422)
```



Comparing above graphs, with theoretically prepared ones – it is clear that they are matching exactly (even after hundreds of frames have been captured), which serves to prove that the system design is robust, and free of bugs. Please note that the theoretical graphs have performed analysis on 10Hz mode, and not 1Hz mode. This was deemed necessary since the 1Hz mode was having way too much difference between deadlines and actual capacities, and also, that if 10Hz mode worked properly – 1Hz mode would work anyway.

Hand Drawn Theoretical RMA

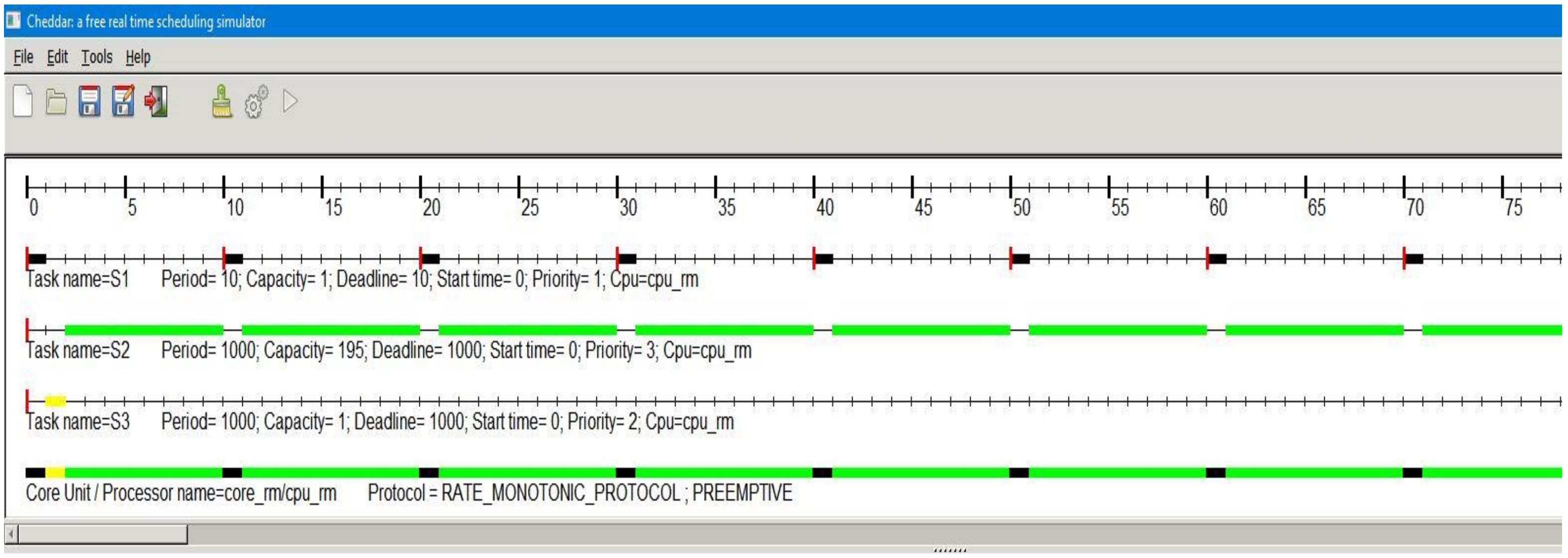


Cheddar RMA

S1 – Scheduler Service

S3 – Cam_Monitor Service

S2 – Cam_RGB Service



Task name=S1 Period= 10; Capacity= 1; Deadline= 10; Start time= 0; Priority= 1; Cpu=cpu_rm

Task name=S2 Period= 1000; Capacity= 195; Deadline= 1000; Start time= 0; Priority= 3; Cpu=cpu_rm

Task name=S3 Period= 1000; Capacity= 1; Deadline= 1000; Start time= 0; Priority= 2; Cpu=cpu_rm

Core Unit / Processor name=core_rm/cpu_rm Protocol = RATE_MONOTONIC_PROTOCOL ; PREEMPTIVE

```
Scheduling simulation, Processor cpu_rm :
- Number of context switches :  45
- Number of preemptions :  21

- Task response time computed from simulation :
   S1 => 1/worst
   S2 => 218/worst
   S3 => 2/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.
```