1. Elaine Cole, Nayan Saikia

2. Operation task function performed in order to generate compute-bound workload: call clock\_gettime() as many times as task’s iteration variable.

Posix provides timer\_create() for high resolution user space timers. However, since we need to find time difference, timer\_create() period expiry did not quite lend itself for our purpose. So, the user space timer is implemented using nanosleep. If period is 0, the workload function will be repeatedly called. If period is some numeric value, timestamp before and after the workload function is taken. If the time diff is less than period, the process will sleep for the diff before invoking the function again.

3. Ways in which implementing periodic task in user space differs from implementing one in kernel space: kernel space requires a timer\_expiration function,

In the init thread, hrtimer is started with 0 to wake up task immediately regardless of period. Once in the task thread, if period is zero, hrtimer is again started with time 0. If however period is non-zero, the diff between timestamp before and after loop is used as time interval for hrtimer start.