Jesús Castillo Benito
21/10/2024

02 - Alarms benchmark

b)

For benchmarking our alarm program to measure its performance against the multithreaded version, I propose the following benchmark.

The proposed benchmark will measure the precision of the alarms and the overhead caused by the program. It will be based on comparing the sum on the absolute difference between the desired time of the alarm and the time when it actually gets triggered.

In addition, if we wanted to do it more precisely, we could also evaluate the CPU and RAM memory consumption, using some linux monitor as for example ‘top’.

To execute this benchmark, we can prepare different sets of alarm times to simulate various workloads. Some examples would be mixing long intervals with short ones (2 seconds for long intervals and 0.5 for short intervals) or creating a random set of alarms with various time intervals (short, long and medium).
In addition, we could modify our program to give an output more appropriate for storing all of it in a file for an easier analysis later on.
For our workloads, we can also modify the number of alarms in each workload, like for example 100, 1000, or even 10000 alarms per workload.

To choose which implementation works better, we evaluate the following factors:

- Sum of Absolute Differences: the lower the sum , the more precise the implementation.
- Latency: measure the time taken from the start of the program until all the alarms are triggered.