Hartstone Hard Real Time Benchmark

Test series, result format and means of comparison specifications

Daniel Casini, Emiliano Palermiti, Matteo Pampana

Abstract This document presents the specification of the test series for the Hartstone Hard Real Time Benchmark, based on [Weidermann 89] and [Donohoe Shapiro Weiderman 90].This is a revisited version of the benchmark, implemented in C (instead of Ada) on two different MCUs, the dsPIC33FJ256MC710 (Evidence FlexBoard) and the STM32F407VG (ST Microelectronics STM32F4-Discovery Board), with two different Real Time Operating Systems (RTOS) such as Erika RTOS and FreeRTOS.

1 Test series specifications

In this section, the test series specification will be presented. In particular will be defined the base structure of the task set whereas the list of the experiments, that has been developed in order to stress the test until a task misses a deadline, will be presented in a next chapter, after having been described the metrics used by the benchmark.

1.1 PH Series

The Periodic Harmonic(PH) is the first of the five test series defined in [Weiderman 89] and implemented in Ada in [Donohoe Shapiro Weiderman 90]. The five periodic tasks are independent: they do not communicate with each other. Each periodic task is characterized by a frequency, a workload, and a priority proportional to the period (rate monotonic). The task frequencies are harmonic: this means that the frequency of a task is an integral mutiple frequency of any lower-frequency task. Frequencies are expressed in Hertz; the reciprocal of the frequency is a task's period. A task workload is a fixed amount of work, which must be completed within a task's period. This workload is expressed as a number of iterations of a variant of the Whetstone Benchmark procedure, which executes a one thousand of Whetstone instructions (Kilo-Whetstone). Hence, an Hartstone task is required to execute a specific number of Kilo-Whetstones per period. The rate at which it does this amount of work is measure in KWIPS (Kilo-Whetstone instructions per second). The workload rate of a task is equal to its workload multipled by the task's frequency. The deadline for completion of the workload is the beginning of the task's next period.

The PH series consists in four experiments that start with a baseline task set and modify some parameters with the aim to increase the ”weight” of the task set, until a deadline is missed.

2 Hartstone Metrics

In this section, the metrics used in the benchmark will be presented.

- **Raw Speed**: At the beginning, the *raw speed* is computed, that is the number of Kilo-Whetstone per seconds (KWIPS) achived by the Whetstone procedure. The *raw speed* can be considered as a metric against which the performance of the task set can be measured: this is because it does not include the non task-workload (context-switch overhead).
- **Percentage of workload utilization (per task)**: Expresses the the performance requested of a Hartstone task. It is computed as the ratio of the requested task

speed to the raw benchmark speed, assuming the raw speed to be 100% utilization.

- **Percentage of workload utilization (global)**: Is the sum of the workload utilization for all the tasks.
- **Step Size**: The *Step Size* of an experiment is an indication of the extra work required of the task set when the next test in an experiment is derived from the current test. It is expressed as a percentage of the raw speed. The *Step Size*e is the granularity, or resolution, of an experiment.
- **Deadline Miss**: The first deadline miss due to an overrun (w.r.t. the task period) is called *Deadline Miss*.
- **Deadline Skipped**: The subsequent deadline misses due to an overrun (i.e., if a task with period 10 takes 30 to execute, it should be possibile to observe 1 deadline miss and 2 deadline skip) are called *Deadline Skip*

3 PH Series Experiments

3.1 Experiment 1

This experiment consist in increasing the frequency of the highest frequency task (Task 5) for each new test, until a deadline is missed. Since the increment is performed in the highest priority task (RM scheduling), the priority remains the same. The frequencies of the other tasks and he per-period workloads of all tasks do not change. The amount by which the frequency increases must preserve the harmonic nature of the task frequencies: this means a minimum increase by an amount equal to the frequency of Task 4. Experiment 1 tests the system's ability to handle fine granularity of time and to switch rapidly between processes(clock resolution and task switching overhead). It is important to consider a Task 5 frequency that is quite far to the clock resolution: the baseline task set parameters must be created considering this issues.

3.1.1 Experiment 2

Experiment 2 starts with the baseline task set and then increases all the frequencies by 1.1, then 1.2, then 1.3, and so on for each new test until a deadline is missed. The per-period workload of all tasks do not change. This scaling preserves the harmonic frequencies. By contrast with Experiment 1, the increasing rates of work affect all tasks, not just one. If Experiment 1 has already been executed, it is necessary to reset the task set parameters to the original one, in order to avoid that Task5 starts with a modified frequency.

3.1.2 Experiment 3

In Experiment 3, differently from Experiment 1 and 2 that acts on the frequencies, the increment is performed on the tasks' workload. In particular, the workload of each task is increased by a certain amount of Kilo-Whetstone per period, until a task misses a deadline. The task frequencies do not change. In order to avoid problems due to the fact that some task frequencies can have been reached values near to the clock resolution in the previous experiments, it is necessary to reset the task parameters. Experiment 3 increases the total workload rate without increasing the system overhead as in the previous experiment. Depending on the raw speed of the benchmark, Experiment 3 may have a large step size. Since a number of Kilo-Whetstone is added to the workload of each task, if the raw speed is low there will be a significant increase in the experiment step size. The number of Kilo-Whetstone added in each test has to be tuned in order to achieve the desired step size, that allows to determine precisely where the breakdown point is.

3.1.3 Experiment 4

In the last experiment, starting with the baseline task set, for each test a new task with the same charateristics (workload, frequency) of Task 3 is added, until a task misses a deadline. The frequencies and workloads of the baseline task set do not change. This sequence tests the system's ability to handle a large number of tasks. Experiment 4 adds both overhead and Kilo-Whetstone per second to the executing task set. It also increases the amount of blocking for Task 4 and 5, that has a priority lower than all the other tasks, including the new ones, that executes with the priority of Task 3.

4 Results Format

4.1 Test characteristics

In each test (for each experiment), a table will be printed with the following format:

- Task ID
- Frequency
- KiloWhetstone Instruction per Period(KWIPP)
- KiloWhetstone Instruction per Seconds(KWIPS)
- Utilization(%)

4.2 Test results

In each test (for each experiment), a table will be printed with the following format:

- Task ID
- Period
- Number of deadlines met (Met)
- Number of deadlines missed (Miss)
- Number of deadlines skipped (Skip)

It is the result of the benchmark, and it is also useful to establish the breakdown utilization (i.e., the utilization at which deadlines are first missed).