PRELIMINARY PERFORMANCE EVALUATION OF HEF SCHEDULING ALGORITHM

Carlos A. Rincón C. and Albert M. K. Cheng
University of Houston
Real-time Systems Lab
carincon@uh.edu and cheng@cs.uh.edu

OUTLINE

- Motivation
- Related work
- Methods and Results
- Conclusion and future work

MOTIVATION

The purpose of the research is to measure the performance of the highest entropy first scheduling algorithm in order to have a guideline about the behavior of the studied algorithm under certain conditions.

The contributions of this paper are:

- Generate multiple task sets by implementing the programs from the SNU real-time benchmark [1] in Wind River Work-bench 3.3 [2] to calculate the WCET. Also, derive the periods by using a linear programming solution aiming to maximize the utilization of the system based on a predefined hyper-period.

- Measure the performance of the HEF algorithm to schedule real-time tasks using as metrics the number of context switches and deadline-miss ratio.

RELATED WORK

ENTROPY AND REAL-TIME SYSTEMS

Rincon and Cheng [3] proposed the mathematical background for using entropy as a parameter to schedule real-time tasks:

Total Entropy of a Task: $H_{Task} = \frac{hperiod}{T_i} * C_i * H_{SU} = log_2(hperiod) * \frac{C_i}{T_i} bits$

(*H_{SU}=Entropy of a single unit of time of the scheduling diagram).*

Total Normalized Entropy of a Task: $NH_{Task} = \frac{H_{Task}}{C_i} bits$

Total Entropy of the System: $H_{Sys} = \sum_{i=1}^{m} H_{Task_i} = log_2(hperiod) * \sum_{i=1}^{m} \frac{C_i}{T_i} bits$

Relationship between HSys and Utilization: we know that the maximum value of entropy is equal to log2(number of possible cases = hperiod), then HSys ≤ log2(hperiod). Based on this inequality we have:

$$log_2(hperiod) * \sum_{i=1}^{m} \frac{C_i}{T_i} \leq log_2(hperiod). \text{ Then } U = \sum_{i=1}^{m} C_i/T_i \leq 1$$

RELATED WORK

HIGHEST ENTROPY FIRST ALGORITHM

1. Determine the schedulability of the given task set using the proposed relationship.
2. Calculate $NHTask$ and $HTask$ for each task.
3. Select the task to be executed using the following criteria:
   a. Select the task with the highest $NHTask$.
   b. If two or more tasks have the highest $NHTask$, then select the task with the highest $HTask$.
   c. If two or more tasks have the highest $HTask$ and one of these tasks is the one running, then select the task that is running (to minimize preemption), else select the task based on its process identifier (PID).
4. Update the values of T_i and C_i for all tasks.
5. Go to step 2 until time = hperiod.

METHODS AND RESULTS

Task set Generation

TABLE I: Selected programs from the SNU real-time benchmark

Task Number	SNU Program	Description
1	sqrt.c	Square root,function implemented by Taylor series
2	fibcall.c	Summing the Fibonacci series
3	crc.c	A demonstration for CRC (Cyclic Redundancy Check) operation
4	minver.c	Matrix inversion for 3x3 floating point matrix
5	select.c	A function to select the Nth largest number in the floating point array size 20

TABLE II: WCET for the selected tasks

Task Number	WCET	ROUND WCET
1	13.34 ms	14 ms
2	7.32 ms	8 ms
3	13.54 ms	14 ms
4	16.41 ms	17 ms
5	25.73 ms	26 ms

TABLE III: Task sets 1 and 2

Task Number	C_i	T_i
1	14 ms	50 ms
2	8 ms	12 ms

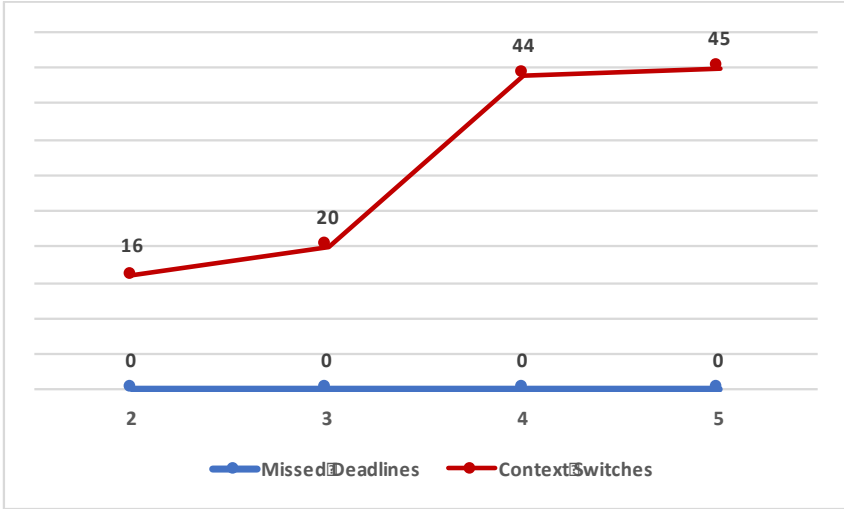
Task Number	C_i	T_i
1	14 ms	100 ms
2	8 ms	50 ms
3	14 ms	20 ms

TABLE IV: Task sets 3 and 4

Task Number	C_i	T_i
1	14 ms	100 ms
2	8 ms	34 ms
3	14 ms	50 ms
4	17 ms	50 ms

Task Number	C_i	T_i
1	14 ms	100 ms
2	8 ms	100 ms
3	14 ms	100 ms
4	17 ms	50 ms
5	26 ms	100 ms

Preliminary Results



CONCLUSIONS AND FUTURE WORK

- We proposed a methodology to generate task sets using the programs from the SNU real-time benchmark (using Wind River Workbench to calculate the WCET of the tasks and a linear programming solution to set the periods).

- The results showed:
  1. The number of context switches is directly proportional to the number of tasks in the task set.
  2. For the deadline-miss ratio, further analysis must be made to confirm that it depends on the utilization of the system ($U \leq 1$ = no deadline misses).

- We propose as future work to compare the performance of HEF against Earliest Deadline First (EDF) [4].

REFERENCES

[1] “Snu real-time benchmark suite,” http://archi.snu.ac.kr/realtime/benchmark.

[2] WindRiver, “Wind river workbench,” http://www.windriver.com.

[3] C. A. Rincon and A. M. Cheng, “Using entropy as a parameter to schedule real-time tasks,” in Real-Time Systems Symposium. WiP Session, 2015 IEEE, Dec 2015, pp. 375–375.

[4] C. Liu and J. Layland, “Scheduling algorithms for multiprogramming in a hard-real-time environment,” J. ACM, vol. 20, no. 1, pp. 46–61, 1973.

QUESTIONS ?

Albert M. K. Cheng

University of Houston – Universidad del Zulia

cheng@cs.uh.edu