CS 330 - Winter 2020
Assignment W4

Due: ~~Wednesday, February 26, 2020~~ Friday, February 28, 2020 (start of class)

You should submit a physical copy of your written homework at the start of class. Be sure to include a collaboration statement with your assignment, even if you worked alone. This is worth 2 points.

[40 points] Problem 1 - Partitioning Heuristics

Consider the following set of implicit-deadline tasks, where task $\tau_i = (T_i, C_i)$.

$$\tau = \{(\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6, \tau_7)\} = \{(4,1),\ (5,1),\ (8,3),\ (10,2),\ (4,2),\ (8,1),\ (12,3)\}.$$

The total system utilization is 1.9, so we hope to schedule these tasks on a two-processor platform.

a) What is an optimal partitioning of these tasks onto two processors, assuming each partition is scheduled using uniprocessor EDF? How do you know each partition is schedulable using EDF?

b) Perform the uniprocessor RM utilization-based test on both of your partitions from part (a). Do the tests pass? Draw the schedule from $t = 0$ to $t = 12$. Does the first job of each task meet its deadline?

c) Use RM-Next-Fit to partition the tasks, assuming the "fit" check is performed using the RM utilization test, and tasks are considered in order of task ID. For full credit, show your work for each fit check. What is the final partitioning, how many processors are required, and what is the utilization of each processor?

d) Given your partitioning result in part (c), what is the ratio m/m_0 between the resulting number of processors m and the optimal number of processors m_0? How does this compare to the bound provided by Dhall and Liu for RM-Next-Fit?

e) Use RM-First-Fit to partition the tasks, assuming the "fit" check is performed using the RM utilization test, and tasks are considered in increasing order of periods (smallest first), with utilization as a tie-breaker (largest first). For full credit, show your work for each fit check. What is the final partitioning, how many processors are required, and what is the utilization of each processor?

f) Given your partitioning result in part (e), what is the ratio m/m_0 between the resulting number of processors m and the optimal number of processors m_0? How does this compare to the bound provided by Dhall and Liu for RM-First-Fit?

g) Use Worst-Fit to partition the tasks between 3 processors, assuming the "fit" check is performed using the RM utilization test, and tasks are considered in decreasing order of utilizations (largest first), with period as tiebreaker (smallest first). For full credit, show your work for each fit check. What is the final partitioning and what is the utilization of each processor?

Recall that Worst-Fit chooses the processor, of those where a task can fit, with the lowest utilization. Assume ties between processors are broken by processor ID (smallest first).

[16 points] Problem 2 - Challenges with Multiprocessor Scheduling

a) Describe the Dhall Effect in your own words.

b) Give an example task system for $m = 4$ that is subject to the Dhall Effect.

c) The critical instant for G-RM no longer corresponds to when all tasks have synchronous releases. Give an example task system for $m = 4$ in which all tasks have synchronous releases, but the worst-case response time for some task does not correspond to its first job.

[18 points] Problem 3 - Processor Sharing and Pfair

a) We discussed a way to visualize a Processor Sharing schedule as a repeated set of frames, where we allocate tasks to processors based on utilizations, and "wrap around" a task to the next processor once one becomes full. Consider again the task set from Problem 1, where task $\tau_i = (T_i, C_i)$.

$$\tau = \{(\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6, \tau_7)\} = \{(4, 1),\ (5, 1),\ (8, 3),\ (10, 2),\ (4, 2),\ (8, 1),\ (12, 3)\}.$$

Draw such a frame allocation for this task set, considering each task in order.

b) Assume the frame size is 0.1 time units. Given your frame depiction in part (a), how much time has task τ_3 executed by time $t = 0.25$? What about by time $t = 4$? How do these compare to the task's utilization?

c) Now assume the frame size is 0.2 time units. Given your frame depiction in part (a), how much time has task τ_1 executed by time $t = 0.5$? What about by time $t = 1$? How do these compare to the task's utilization?

d) Pfair scheduling has no utilization loss. Explain in your own words what this means. Unfortunately, it is impractical to implement. Why is Pfair impractical?

[24 points] Problem 4 - Baker's G-EDF Schedulability Test

a) Consider the following task set, in which each task is represented as $\tau_i = (T_i, C_i, D_i)$:

$$\tau = \{(\tau_1, \tau_2, \tau_3, \tau_4)\} = \{(8, 1, 5), (16, 8, 16), (24, 16, 20), (48, 16, 40)\}.$$

For this task set, to check schedulability under G-EDF using Baker's test, we would have to perform the test (Theorem 12 in [Baker 2003]) for each task τ_k. Show the computation for $k = 1$ assuming $m = 4$. For full points, list each value of β_i and indicate whether the test passes or fails.

b) For the same task set from part (a), perform Baker's test for $k = 3$ assuming $m = 4$. Again, list each value of β_i and show the final comparison for the test.

c) The test from part (b) should have failed. Make one change to a single task parameter that causes the test to pass for $k = 3$ assuming $m = 4$.

d) Rather than test all tasks with your modified task set from part (c), perform the simplified G-EDF test (Corollary 13 in [Baker 2003]) on that task system. What does this tell you about the schedulability of this task system under G-EDF? What about the feasibility of the task system in general on a system with $m = 4$?