Evaluation of Fixed-Priority Scheduling with Deferred Preemption

Abstract
Recently, two generalizations of FPDS (fixed-priority scheduling with deferred preemption) have been proposed: FPDS\(^{\ast}\) [2] and FPDS\(^{\ast\ast}\) [3]. Both generalizations have the potential to reduce the preemption costs of tasks compared to FPDS. The goal of this assignment is to qualify this reduction by means of simulations. The result is expected to contribute to a forthcoming publication.

Description

Background
Fixed-priority scheduling with deferred preemption (FPDS) or cooperative scheduling [1][4] has been proposed as a viable alternative between the extremes of fixed-priority preemptive scheduling (FPPS) and fixed-priority non-preemptive scheduling (FPNS). Compared to FPPS, FPDS (i) reduces memory requirements and (ii) reduces the cost of arbitrary preemptions. Compared to both FPPS and FPNS, FPDS may significantly improve the schedulability ratio of task sets.

For FPDS, each job of a task is assumed to consist of a sequence of sub-jobs, where sub-jobs are non-preemptable. When a task is executing, it can be preempted only between consecutive sub-jobs, i.e. at so-called preemption points. To further improve on FPDS while maintaining the non-preemptive nature of sub-jobs, two generalizations of FPDS have been proposed: FPDS\(^{\ast}\) [2] and FPDS\(^{\ast\ast}\) [3].

For FPDS\(^{\ast}\), each task has a preemption threshold. A task competes for the processor at its priority when it is ready to execute but has not yet started to execute, similar to FPDS. When a task is executing, it can only be preempted at its preemption points by tasks with a priority higher than its preemption threshold.

FPDS\(^{\ast\ast}\) generalizes FPDS\(^{\ast}\) by having a dedicated preemption threshold for each preemption point.

Goal
We are interested in the relative strength of FPDS\(^{\ast}\) and FPDS\(^{\ast\ast}\) compared to FPDS, in particular concerning the cost of preemptions. The goal of this assignment is therefore to qualify the relative cost of preemptions by means of simulations.

Approach
Recently, an algorithm has been described to assign optimal priorities to a set of tasks scheduled by means of FPDS [5]. Unfortunately, similar algorithms do not yet exist for FPDS\(^{\ast}\) and FPDS\(^{\ast\ast}\). For the assignment, we therefore propose the following main steps:

1. Implement the algorithm to determine optimal priorities for FPDS as described in [5].
2. Evaluate the relative strength of FPDS\(^{\ast}\) and FPDS\(^{\ast\ast}\) compared to FPDS for task sets that are schedulable by FPDS.

Because FPDS\(^{\ast}\) and FPDS\(^{\ast\ast}\) both generalize FPDS, we know that under FPDS\(^{\ast}\) and FPDS\(^{\ast\ast}\) the last sub-job of every task under FPDS\(^{\ast}\) and FPDS\(^{\ast\ast}\) can be at least as large as under FPDS. Moreover, for FPDS we can determine the number of preemption points for a task given the length of its last sub-job.
FPDS\textsuperscript{*} is considered to perform “better” for a task set $\Gamma$ than FPDS when:

1. The length of a last sub-job of a task of $\Gamma$ under FPDS\textsuperscript{*} is larger than under FPDS, because that may give rise to less preemption points or provide more freedom in placing preemption points, or
2. The preemption threshold of a task of $\Gamma$ under FPDS\textsuperscript{*} is higher than its priority.

FPDS\textsuperscript{•} is considered to perform “better” for a task set $\Gamma$ than FPDS when:

1. The length of a last sub-job of a task of $\Gamma$ under FPDS\textsuperscript{•} is larger than under FPDS, or
2. The preemption threshold of a preemption point of a task of $\Gamma$ under FPDS\textsuperscript{•} is higher than its priority.

There exists software to generate task sets and to construct optimal configurations of tasks under FPDS\textsuperscript{*} and FPDS\textsuperscript{•}. Checking the 2nd condition for FPDS\textsuperscript{•} may require additional work, however, because there are several ways to construct an optimal FPDS\textsuperscript{*}-configuration.

**Expected results**

The following main results are expected:

a) An implementation of the algorithm to determine an optimal priority assignment for a task set under FPDS as described in [5].

b) An algorithm with an accompanying implementation to construct an FPDS-compliant configuration for FPDS\textsuperscript{*} which allows checking the 2nd condition above.

c) An extensive evaluation.

d) An MSc-report.

**References**


