The Global Limited Preemptive Earliest Deadline First Feasibility of Sporadic Real-time Tasks

Abhilash Thekkilakattil, Sanjoy Baruah, Radu Dobrin and Sasikumar Punnekkat

MÄLARDALEN UNIVERSITY SWEDEN

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Vetenskapsrådet

Motivation

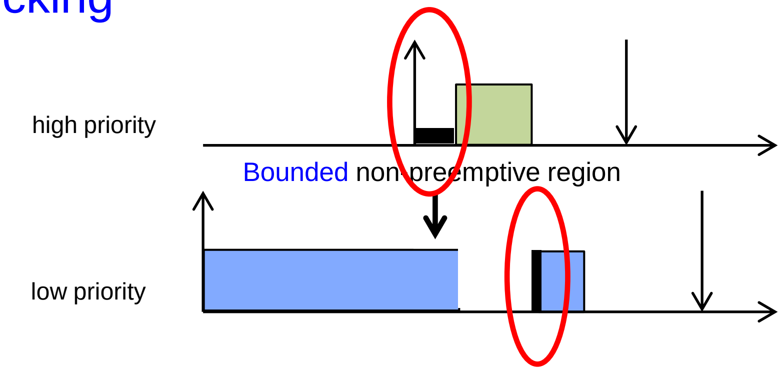
- Multi (-core) processors in real-time systems complicate the problems associated with fully preemptive schedulers
  - Complex hardware, e.g., different levels of caches
    - Difficult to perform timing analysis
  - Potentially large number of task migrations: implementation issues
    - Difficult to demonstrate predictability
    - Difficult to reason about safety
- Non-preemptive scheduling can be infeasible at arbitrarily small utilization
  - Long task problem: at least one task has execution time greater than the shortest deadline

Solution: Limit preemptions

Advantages of limiting preemptions

Combines best of preemptive and non-preemptive scheduling

- Control preemption related overheads
  - Context switch costs, cache related preemption delays, pipeline delays and bus contention costs
- Improve processor utilization
  - Reduce preemption related costs while eliminating infeasibility due to blocking

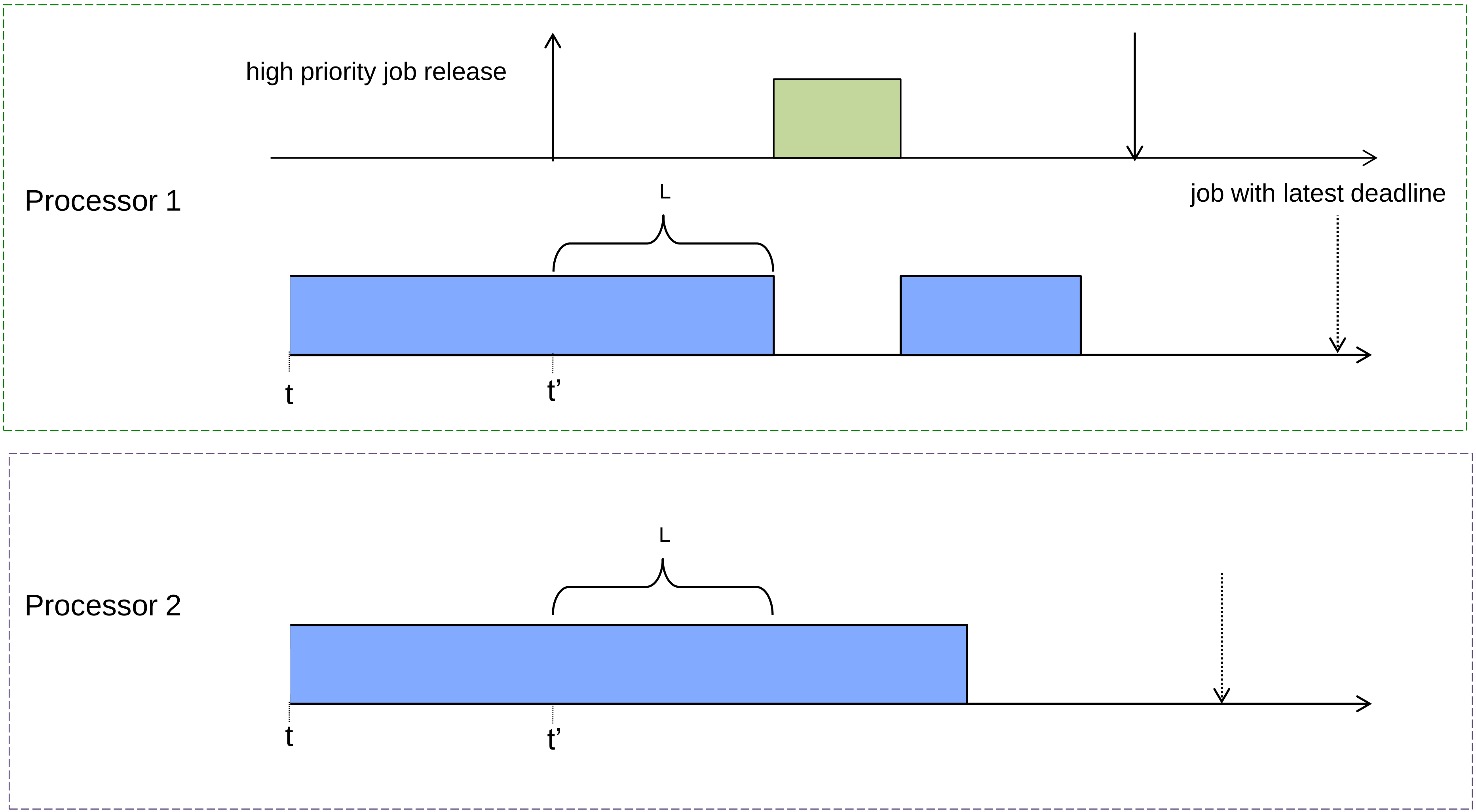


Anecdotal evidence: "*limiting preemptions improves safety and makes it easier to certify software for safety-critical applications*"

Limited preemptive scheduling landscape

Uniprocessor	Limited preemptive FPS (Yao et al., RTSJ'11)	Limited preemptive EDF (Baruah, ECRTS'05)
Multiprocessor	Global limited preemptive FPS (Marinho et al., RTSS'13)	**?**

G-LP-EDF scheduling model



Low priority job executions

High priority job executions

Main contributions

1. Schedulability analysis for Global Limited Preemptive Earliest Deadline First (G-LP-EDF) scheduling of sporadic real-time tasks

2. Analysis of the effects of increasing the processor speed on G-LP-EDF feasibility

Methodology overview

A necessary unschedulability condition:

A sufficient schedulability condition:

Upper-bound on the work generated under G-LP-EDF

$\geq$

$\leq$

Lower-bound on the work executed under any work conserving algorithm

MÄLARDALEN UNIVERSITY SWEDEN

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Methodology overview

A sufficient schedulability condition:

Upper-bound on the work generated under G-LP-EDF

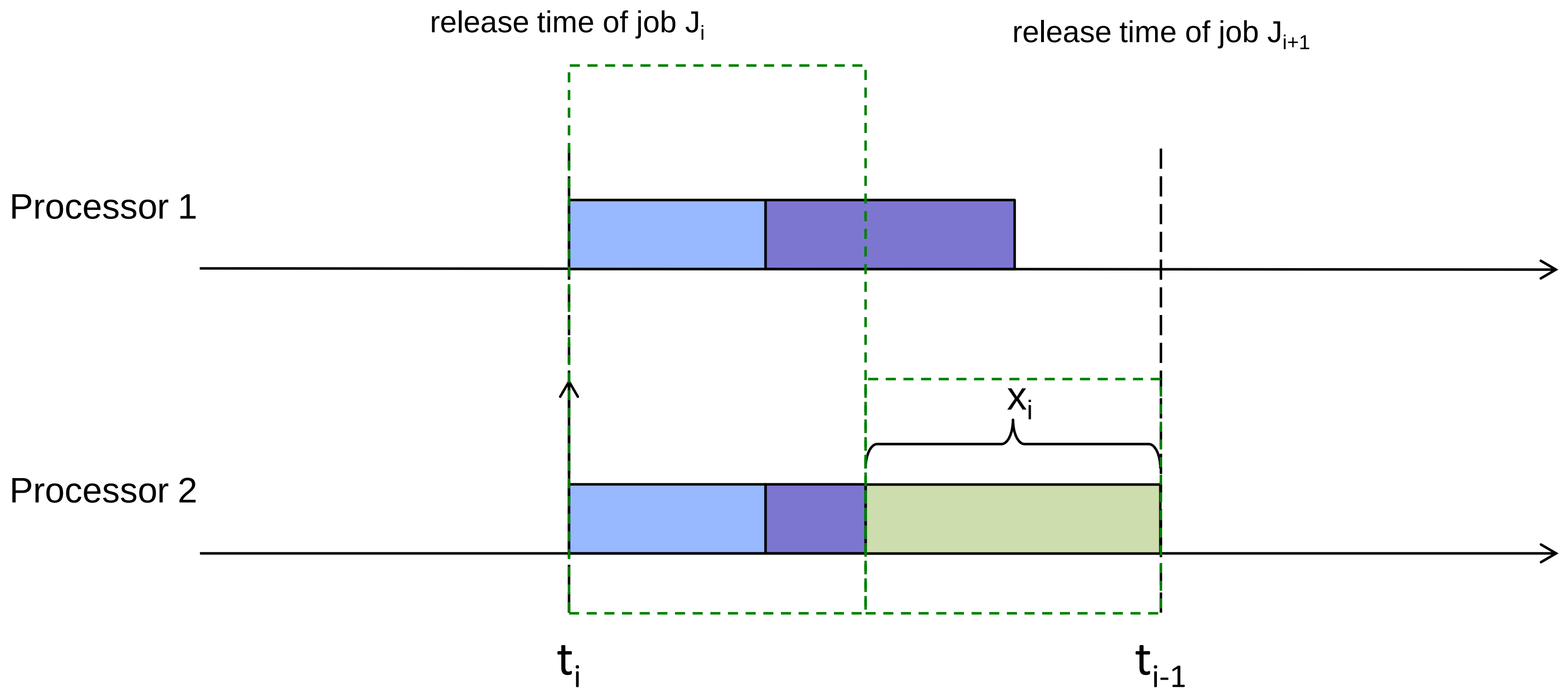
$\leq$

Lower-bound on the work executed under any work conserving algorithm

MÄLARDALEN UNIVERSITY SWEDEN

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Lower bound on the work done



work done(t_i, t_{i-1}) ≥ $m\left(t_{i-1} - t_i - x_i\right) + x_i$

J_i's execution

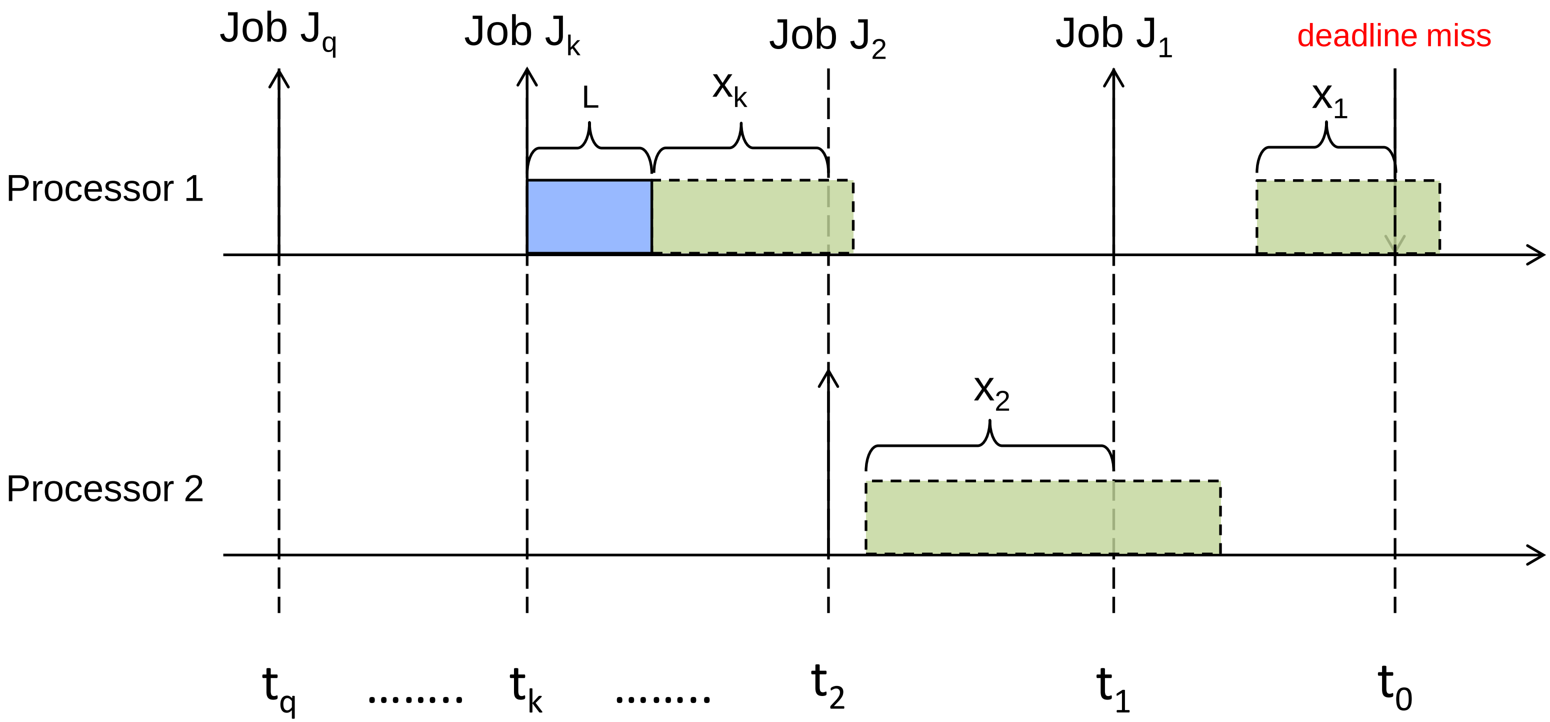
High priority job executions

Low priority job executions (blocking)

Lower bound on the work done

Unschedulability scenario:

$x_i < (t_{i-1} - t_i)s$ $\quad$ $x_1 < (t_0 - t_1 - L)s$ $\quad$ $\max_{\tau_i} \left[\frac{C_i}{D_i - L} \right] \le s \le 1$



work done(t_i, t_{i-1}) ≥ $m\left(t_{i-1} - t_i - x_i\right) + x_i$

work done(t_k, t_0) > $\left(m - (m-1)s\right)\left(t_0 - t_k - L\right) + mL$

Low priority job executions (blocking) $\quad$ High priority job executions

MÄLARDALEN UNIVERSITY SWEDEN

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Methodology overview

A sufficient schedulability condition:

Upper-bound on the work generated under G-LP-EDF

≤

Lower-bound on the work executed under any work conserving algorithm

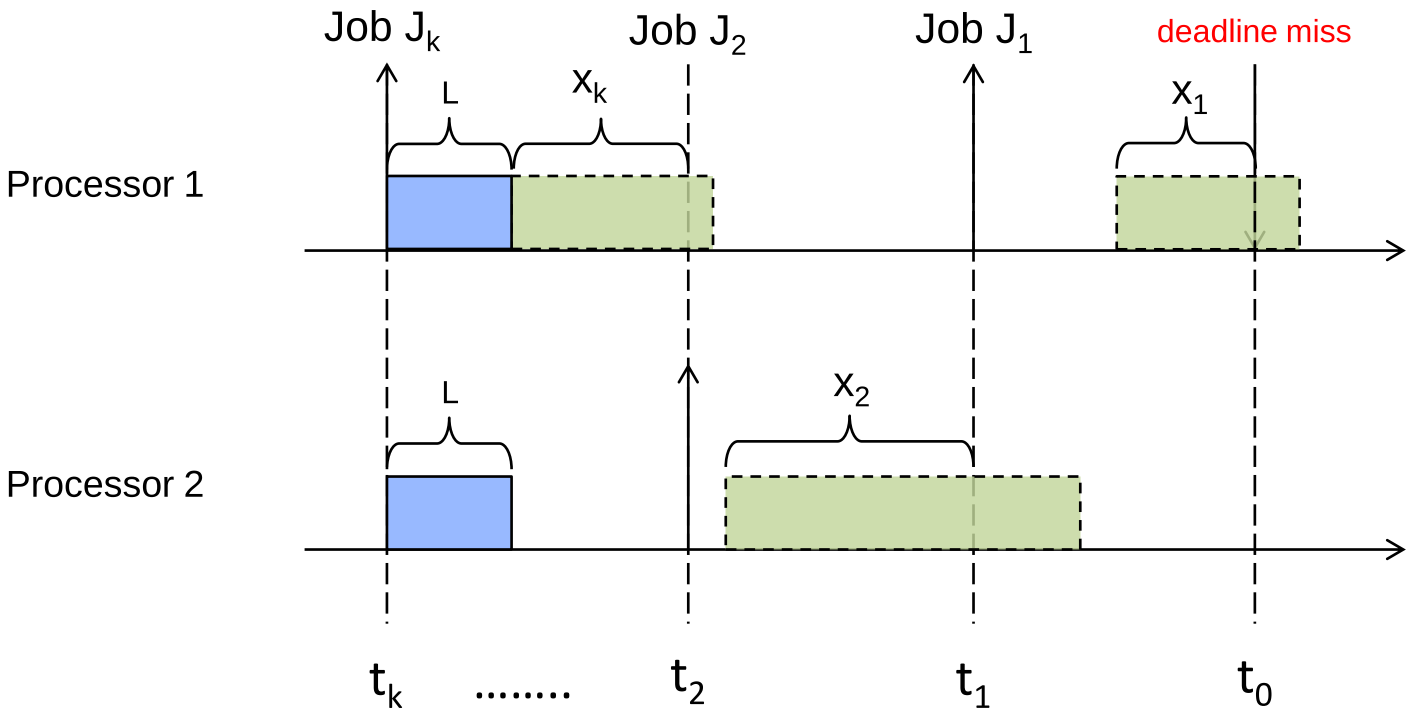
Upper bound on the work generated

….. under G-LP-EDF

In $[t_k,t_0)$, we consider the duration for which:

a. Low priority tasks block high priority tasks
b. Higher priority tasks execute

Maximum duration of blocking



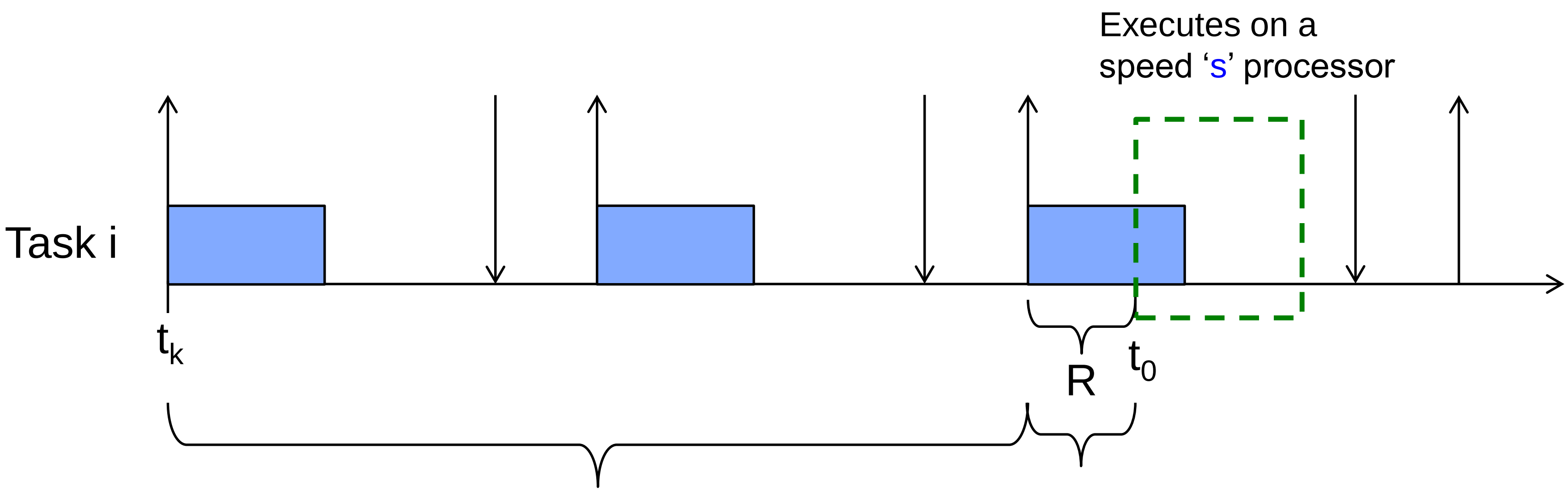
blocking(t_k, t_0) : mL

Low priority job executions (blocking)

High priority job executions

Interference from higher priority tasks / Upper bound on the work done by a task

Work done(t_k,t_0) $\leq \sum_{\forall \tau_i} FF\text{-}DBF_i(t_0 - t_k, s) + mL$



$$FF\text{-}DBF_i(t,s) = head_i + tail_i$$

(Forced-Forward Demand Bound Function)

$$head_i = \left\lfloor \frac{t}{T_i} \right\rfloor C_i$$

$$tail_i = \begin{cases} C_i & R \geq D_i \\ C_i - (D_i - R)s & D - \frac{C_i}{s} \leq R < D_i \\ 0 & \text{otherwise} \end{cases}$$

MÄLARDALEN UNIVERSITY SWEDEN

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Methodology overview

A sufficient schedulability condition:

Upper-bound on the work generated under G-LP-EDF ≤ Lower-bound on the work executed under any work conserving algorithm

$$\sum_{\forall \tau_i} FF\text{-}DBF_i(t_0 - t_k, s) + mL \le (m-(m-1)s)(t_0 - t_k - L) + mL$$

$$\Rightarrow \sum_{\forall \tau_i} FF\text{-}DBF_i(t_0 - t_k, s) \le (m-(m-1)s)(t_0 - t_k - L)$$

MÄLARDALEN UNIVERSITY SWEDEN

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Agenda

1. Schedulability analysis for Global Limited Preemptive Earliest Deadline First (G-LP-EDF) scheduling of sporadic real-time tasks

2. Analysis of the effects of increasing the processor speed on G-LP-EDF feasibility

MÄLARDALEN UNIVERSITY SWEDEN

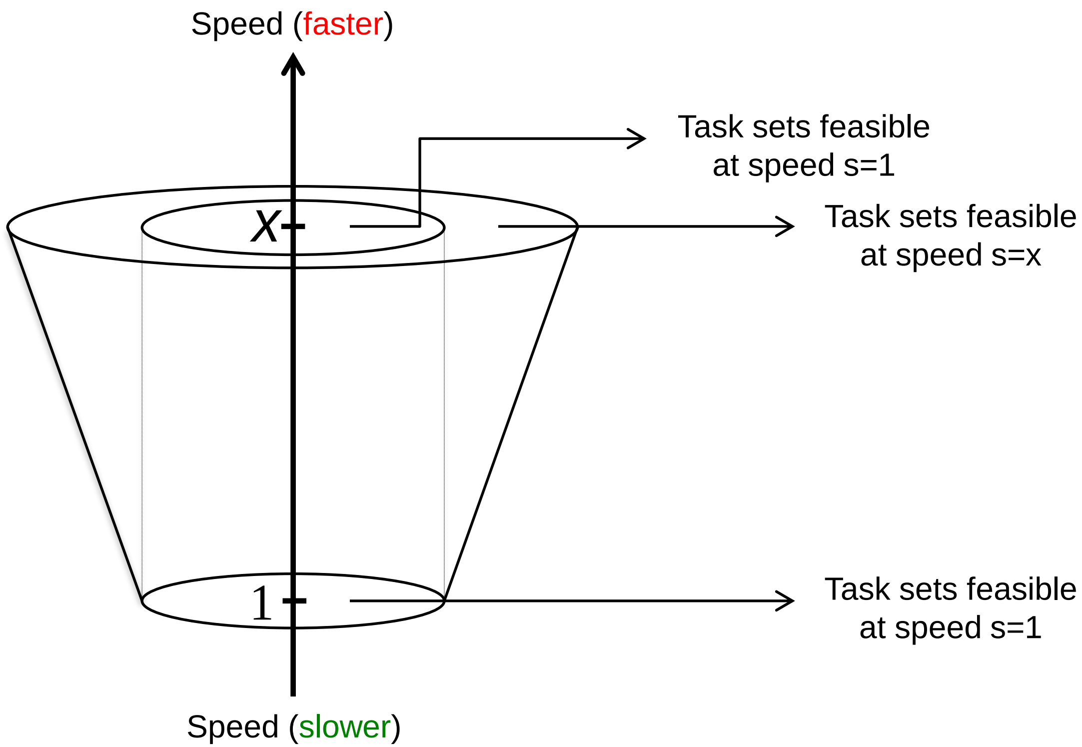
THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Agenda

1. Schedulability analysis for Global Limited Preemptive Earliest Deadline First (G-LP-EDF) scheduling of sporadic real-time tasks

2. Analysis of the effects of increasing the processor speed on G-LP-EDF feasibility

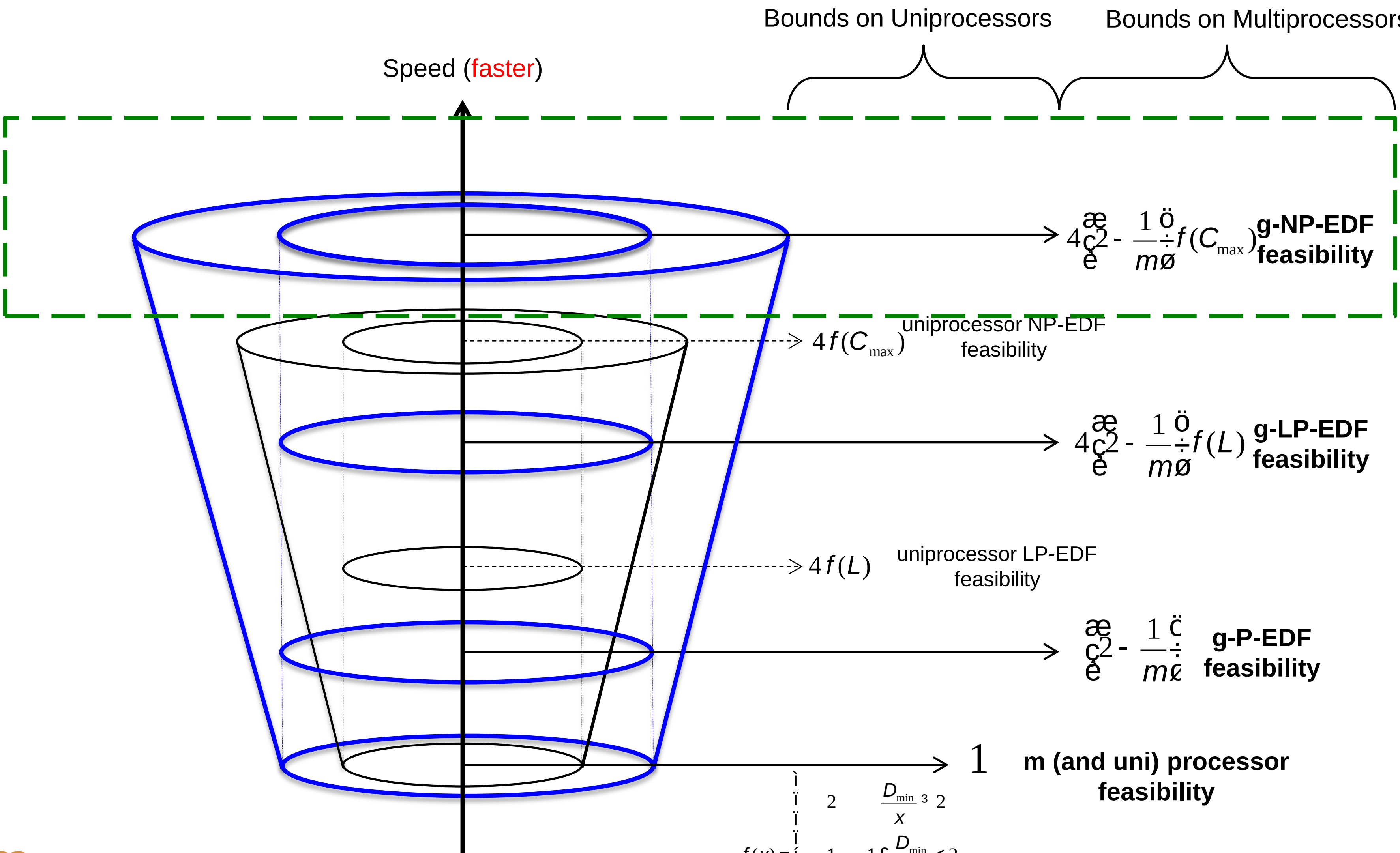
Feasibility bucket



MÄLARDALEN UNIVERSITY SWEDEN

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Processor speed vs. LP-EDF feasibility



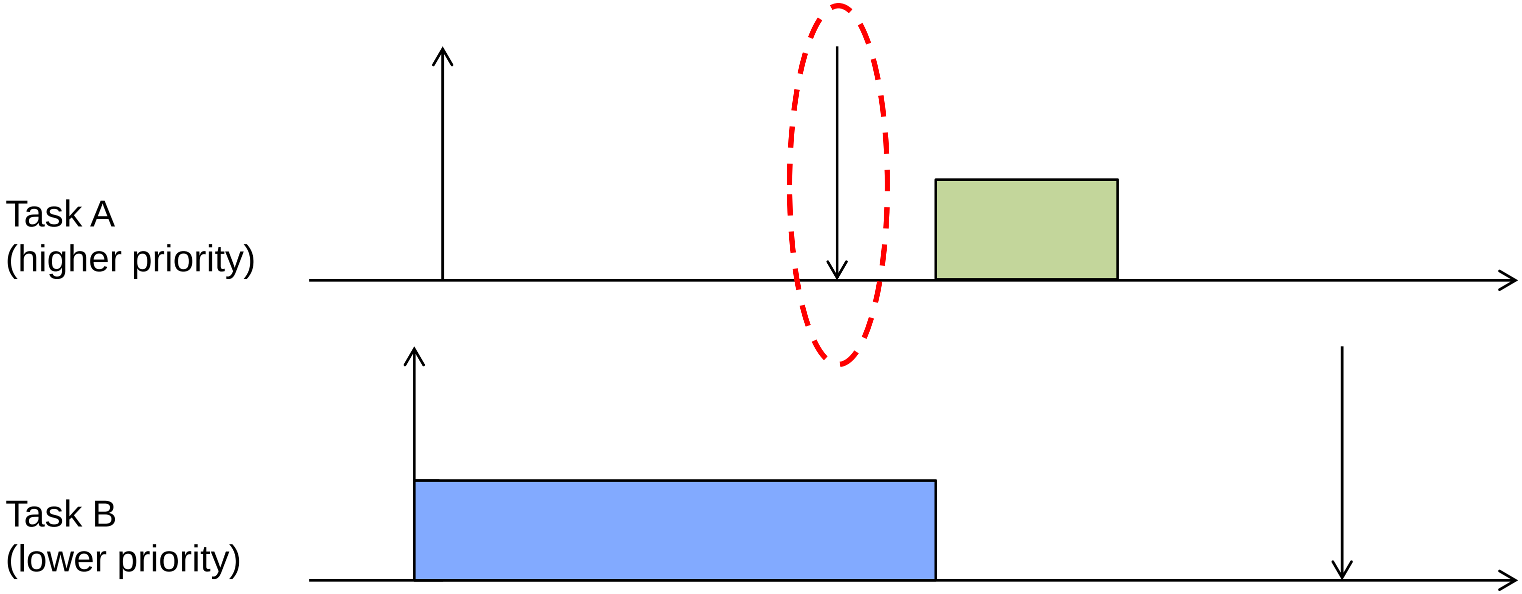
$$f(x)=\begin{cases} 2 & \frac{D_{min}}{x}\geq 2 \\ 1 & 1\leq\frac{D_{min}}{x}<2 \\ \frac{x}{D_{min}} & 0<\frac{D_{min}}{x}<1 \end{cases}$$

MÄLARDALEN UNIVERSITY SWEDEN

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Long task problem

Non-preemptive infeasibility arising from at least one task having WCET greater than shortest deadline



A solution: code-refactoring at task level

Our speed-up factor quantifies the extent to which code-refactoring must be done to enable non-preemptive feasibility

(Short, 2010, "The case for non-preemptive, deadline-driven scheduling in real-time embedded systems")

Agenda

1. Schedulability analysis for Global Limited Preemptive Earliest Deadline First (G-LP-EDF) scheduling of sporadic real-time tasks
2. Analyze the effects of increasing the processor speed on G-LP-EDF feasibility

MÄLARDALEN UNIVERSITY SWEDEN

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Conclusions

- Global limited preemptive EDF feasibility analysis
  - To control preemption related overheads
  - Enables better reasoning about predictability of multi (-core) processor real-time systems
- Processor speed vs. preemptive behavior
  - Quantifies the extent to which code-refactoring must be performed to address the long task problem
  - Sub-optimality of G-NP-EDF

Future work

- Compare G-LP-EDF and G-P-EDF in presence of overheads
- Perform trade-offs: number of extra processors vs. speed-up
- Partition tasks comprising of non-preemptive chunks
- Accounting for suspensions

Thank you !

Questions ?

MÄLARDALEN UNIVERSITY SWEDEN

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL