Multiprocessor Fixed Priority Scheduling with Limited Preemptions

Abhilash Thekkilakattil, Rob Davis, Radu Dobrin, Sasikumar Punnekkat and Marko Bertogna

MÄLARDALEN UNIVERSITY SWEDEN

UNIVERSITY of York

BITS Pilani
K K Birla Goa Campus

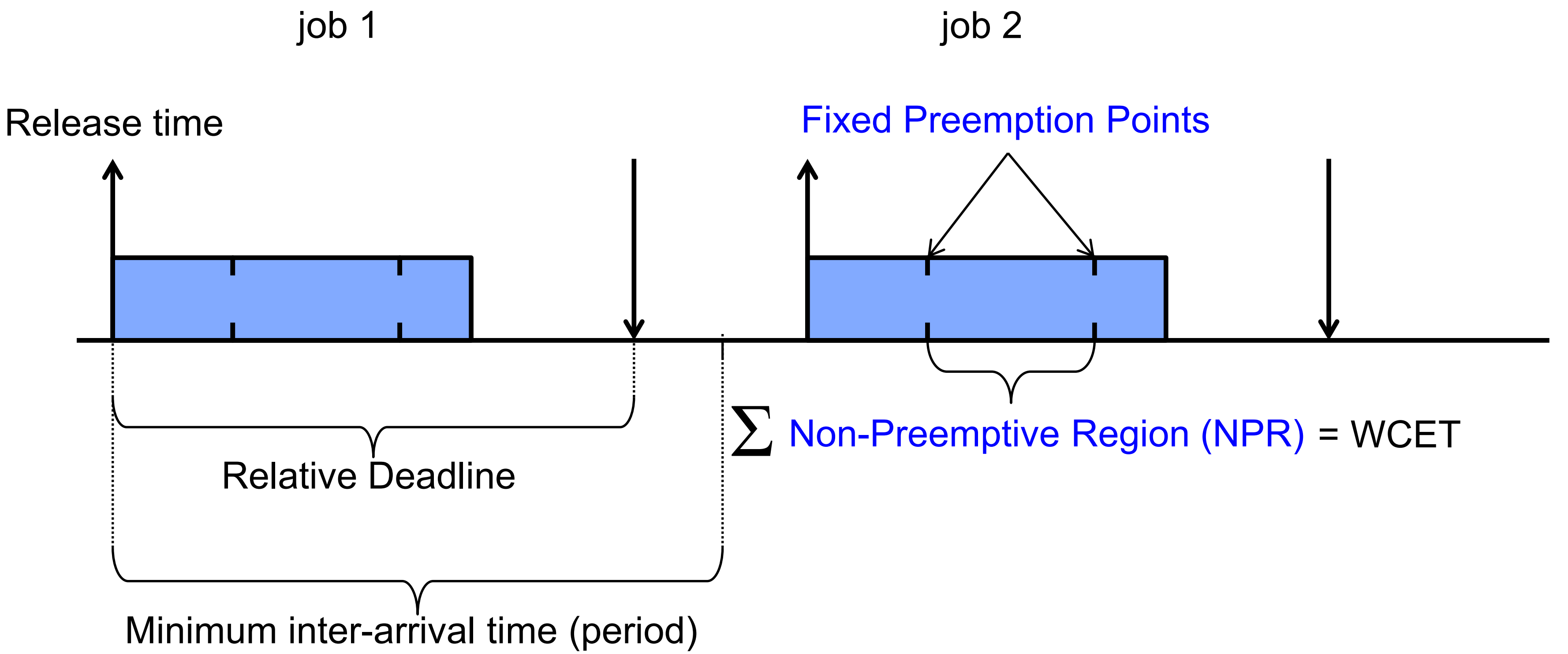
UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Motivation

- Preemptive scheduling on multi (-core) processors introduces new challenges
  - Complex hardware, *e.g.,* different levels of caches
    - Difficult to perform timing analysis
  - Potentially large number of task migrations
    - 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

One solution: limit preemptions

System Model

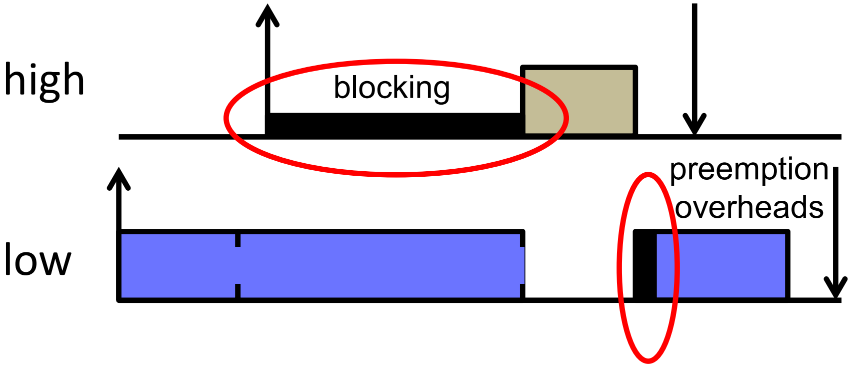


Identical multiprocessor platform with *m* processors

Limited Preemptive Scheduling

Combines best of preemptive and non-preemptive scheduling

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



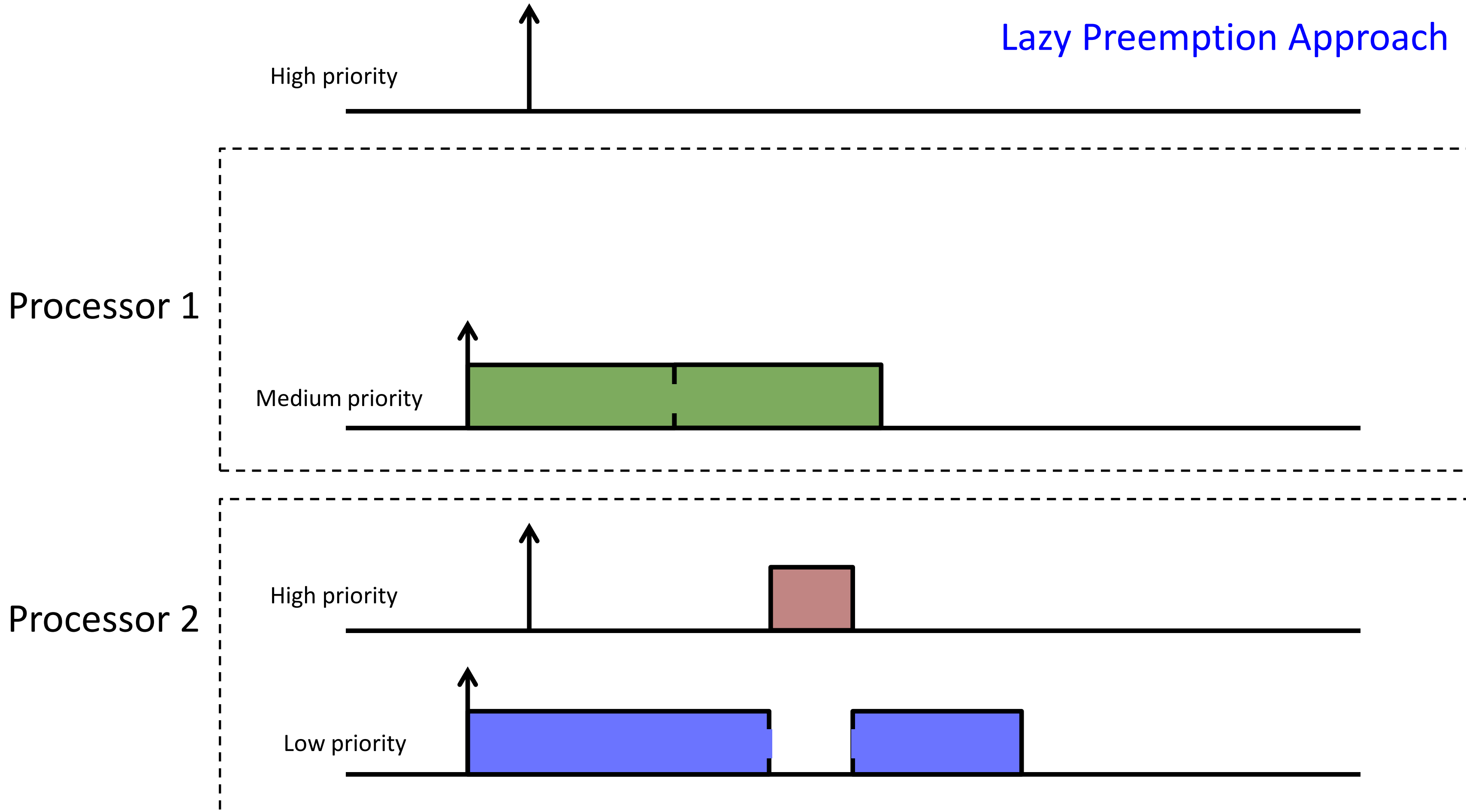
Anecdotal evidence: "*limiting preemptions improves safety and makes it easier to certify software for safety-critical applications*"

Limited preemptive scheduling landscape

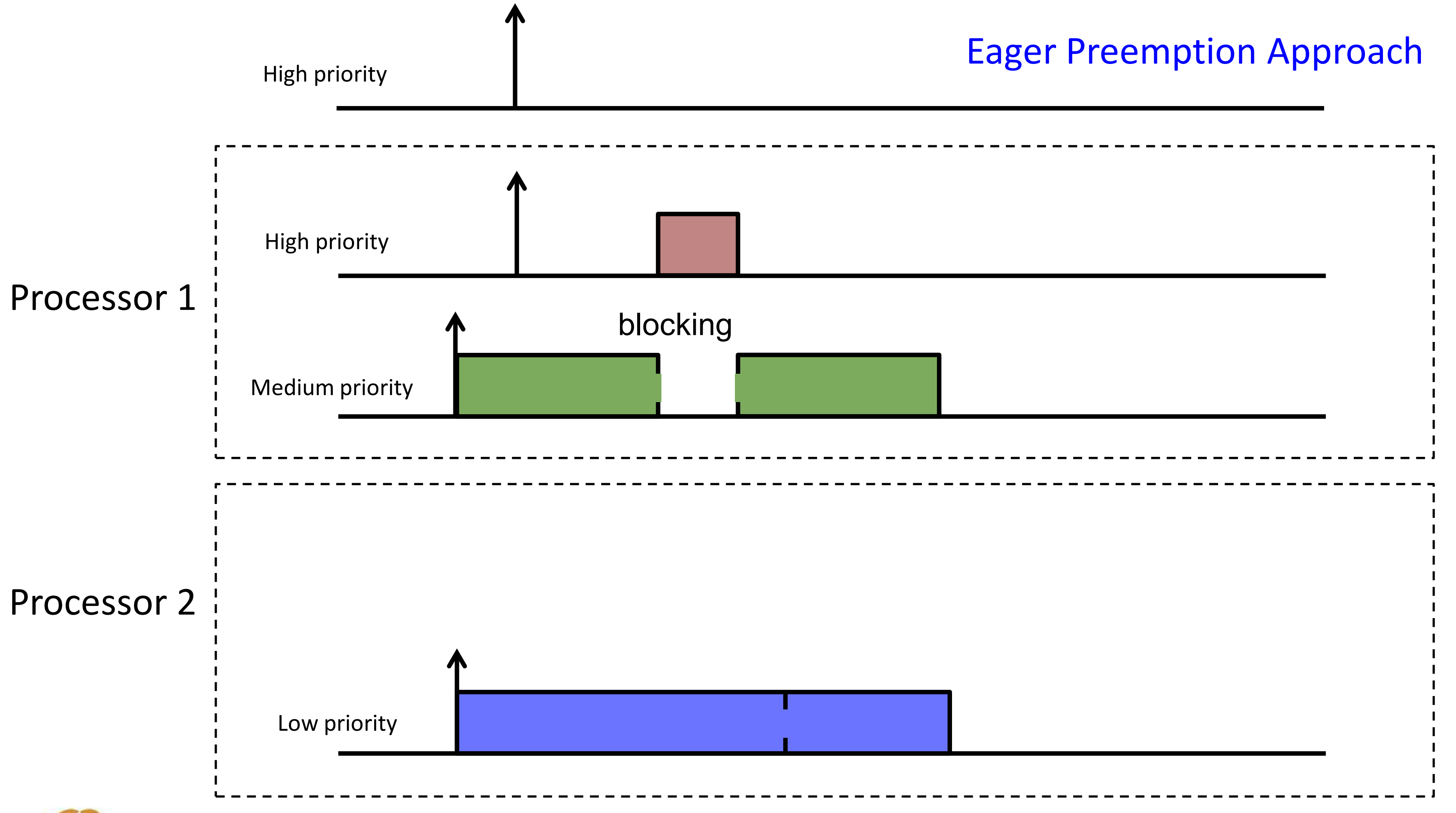
Uniprocessor	Limited preemptive FPS (Burns'94, Bril *et al.*, RTSJ'09, Yao *et al.*, RTSJ'11)	Limited preemptive EDF (Baruah, ECRTS'05)
Multiprocessor	Global limited preemptive FPS (Block *et al.*, RTCSA'07, Marinho *et al.*, RTSS'13, Davis *et al.*, TECS'15)	Global limited preemptive EDF (Block *et al.*, RTCSA'07, Thekkilakattil *et al.*, ECRTS'14, Chattopadhyay and Baruah, RTNS'14)

… of course the references are by no way exhaustive!

Managing Preemptions in Global Limited Preemptive Scheduling



Managing Preemptions in Global Limited Preemptive Scheduling



Global Limited Preemptive FPS with Fixed Preemption Points

Lazy Preemption Approach	Block *et al.*, RTCSA'07: Link Based Scheduling
Eager Preemption Approach	

Lazy Preemption Approach: Link Based Scheduling

- Developed in the context of resource sharing by Block *et al.*, RTCSA'07
  - Applicable to limited preemptive scheduling

- Implements lazy preemption approach

- Higher priority tasks blocked on a processor is linked to that processor

- Analyzable using a simple and generic inflation based test (Brandenburg and Anderson, MPI-Tech Report'14)

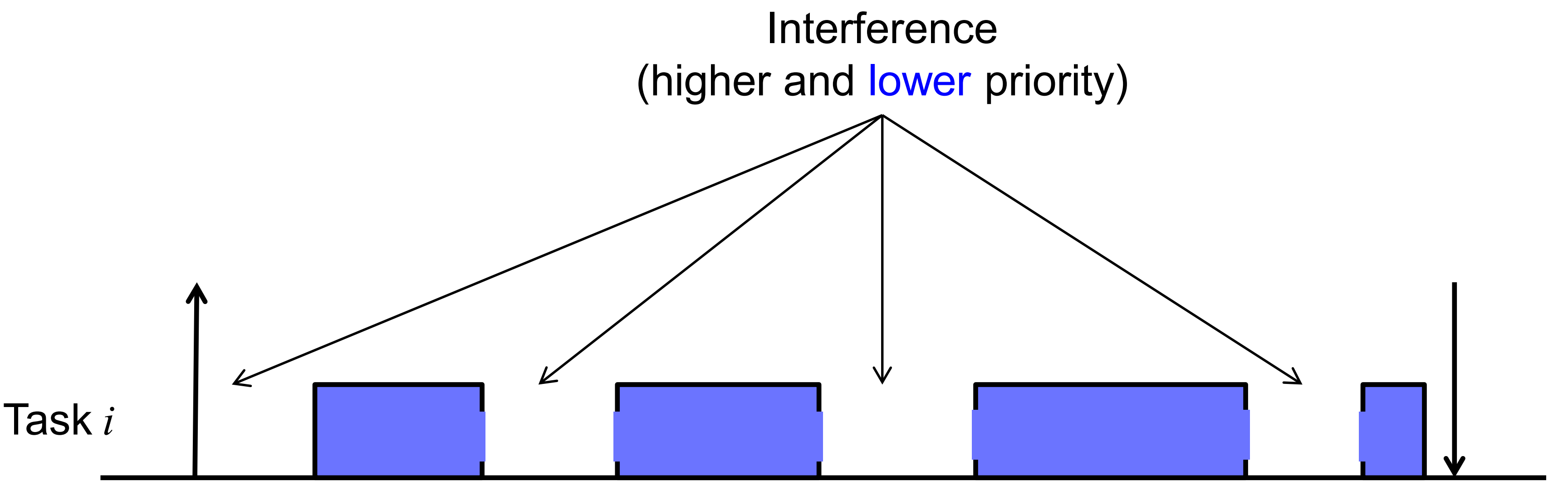
  1) Inflate WCET with largest blocking factor
  2) Determine schedulability using any standard test *e.g.,* response time analysis for global preemptive FPS

Global Limited Preemptive FPS with Fixed Preemption Points

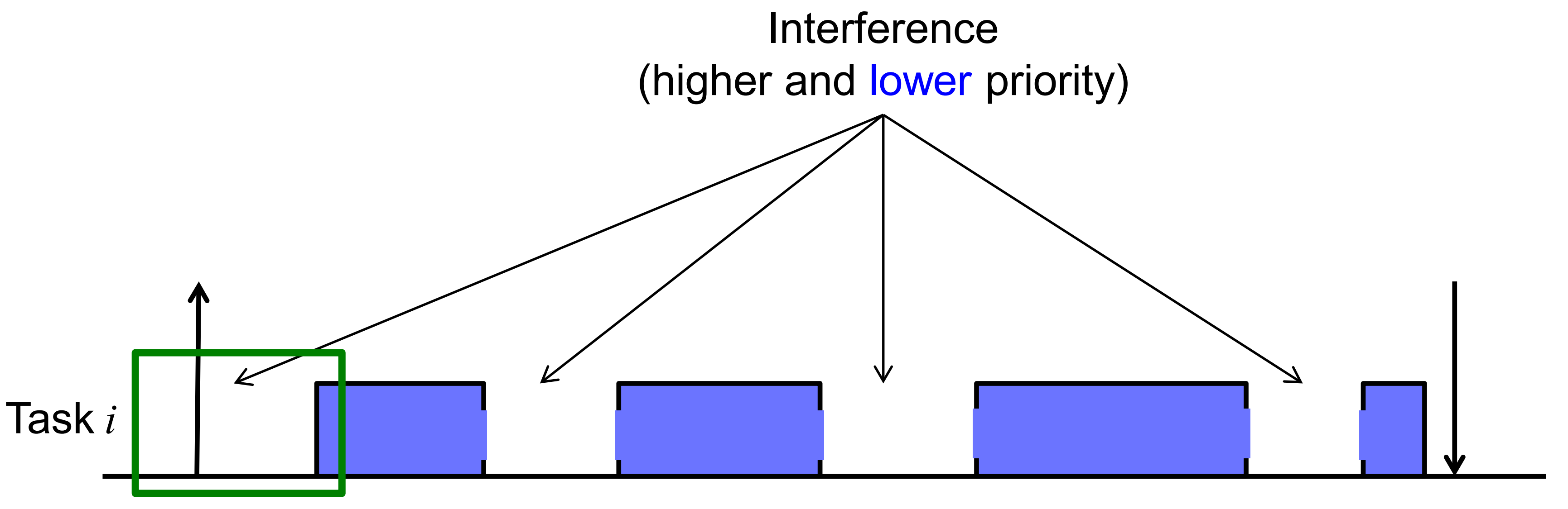
Lazy Preemption Approach	Block *et al.*, RTCSA'07: Link Based Scheduling
Eager Preemption Approach	No significant work!

How can we perform schedulability analysis of tasks scheduled using G-LP-FPS with eager preemptions?

Schedulability Analysis under G-LP-FPS with Eager Preemptions

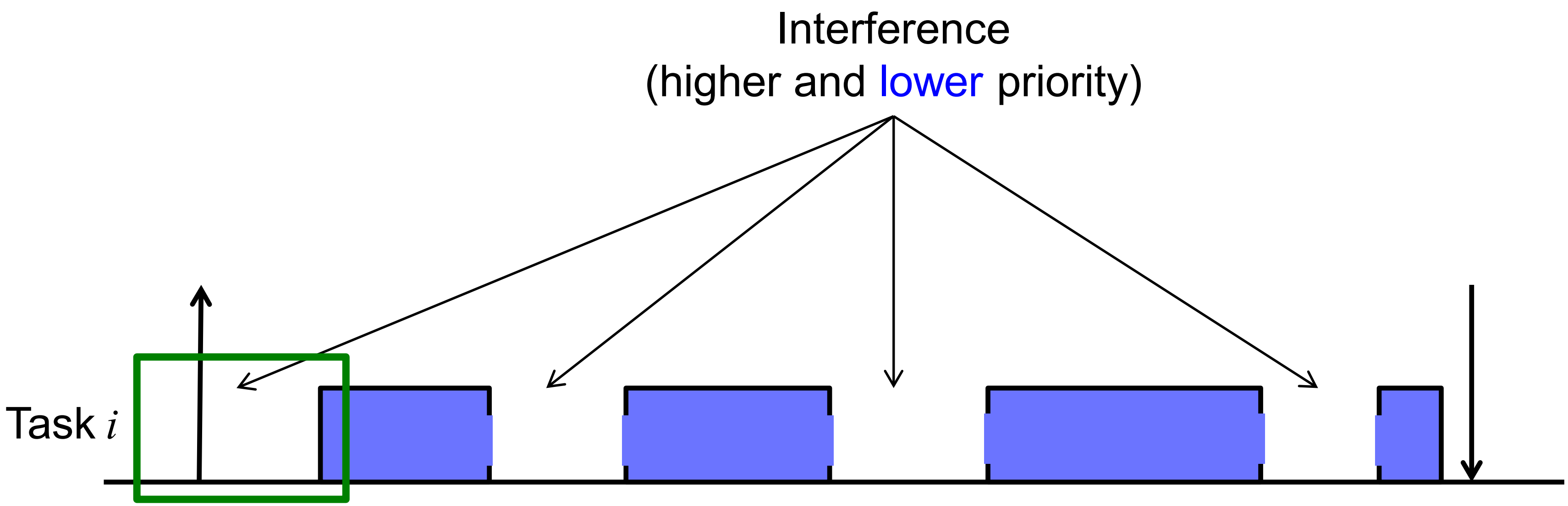


Schedulability Analysis under G-LP-FPS with Eager Preemptions



- Case 1: no “push through” blocking
- Case 2: presence of “push through” blocking

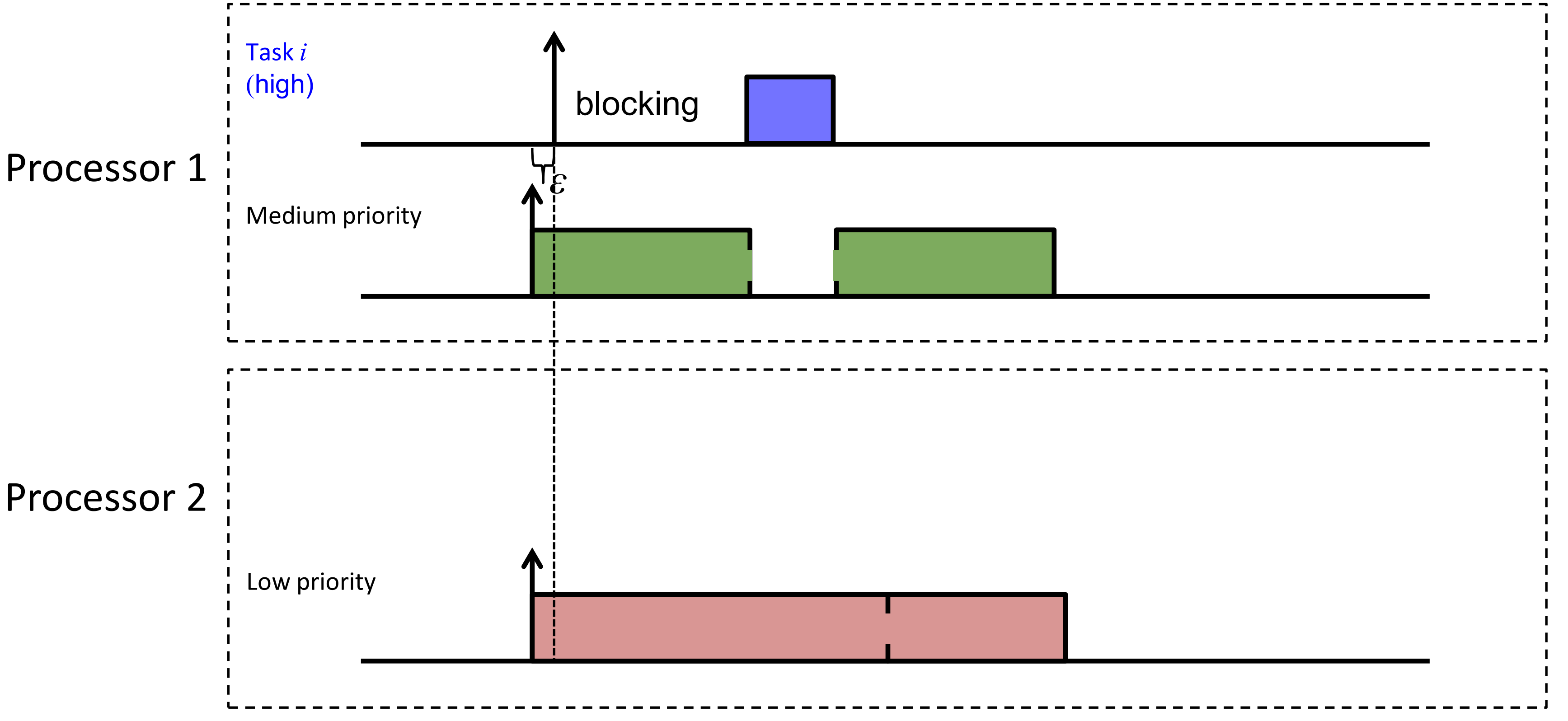
Schedulability Analysis under G-LP-FPS with Eager Preemptions



- Case 1: no “push through” blocking
- Case 2: presence of “push through” blocking

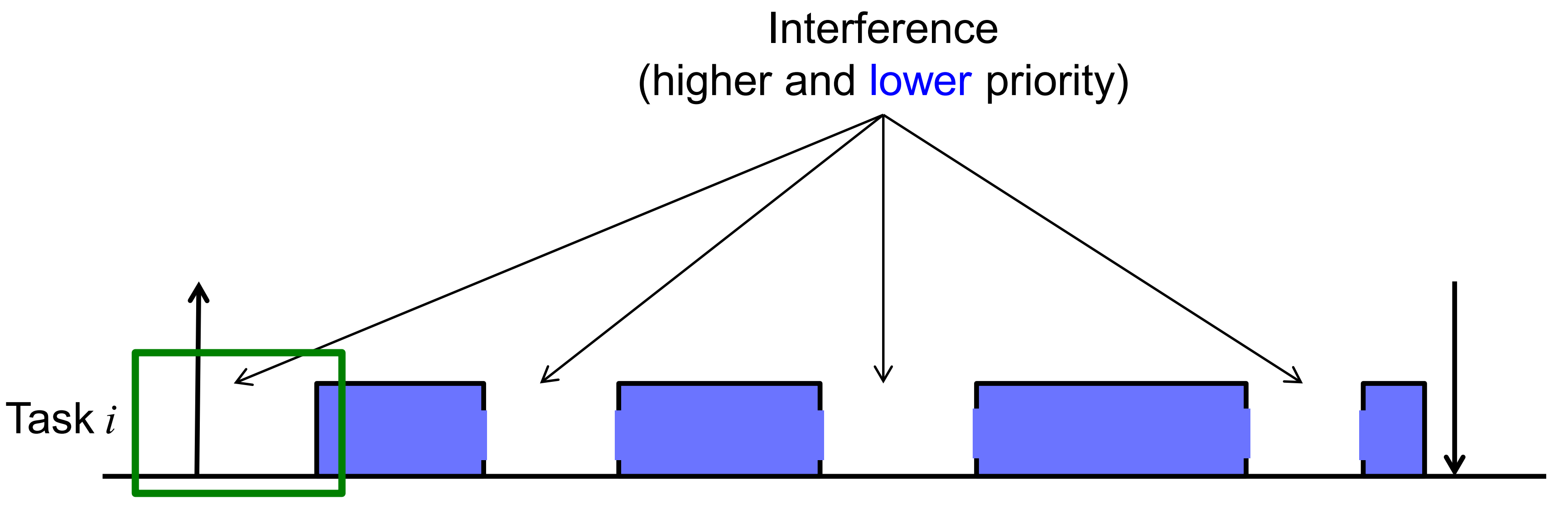
Lower Priority Interference before Task Start Time

Case 1: no push through blocking



blocking= sum of *m* largest ({lower priority NPRs})

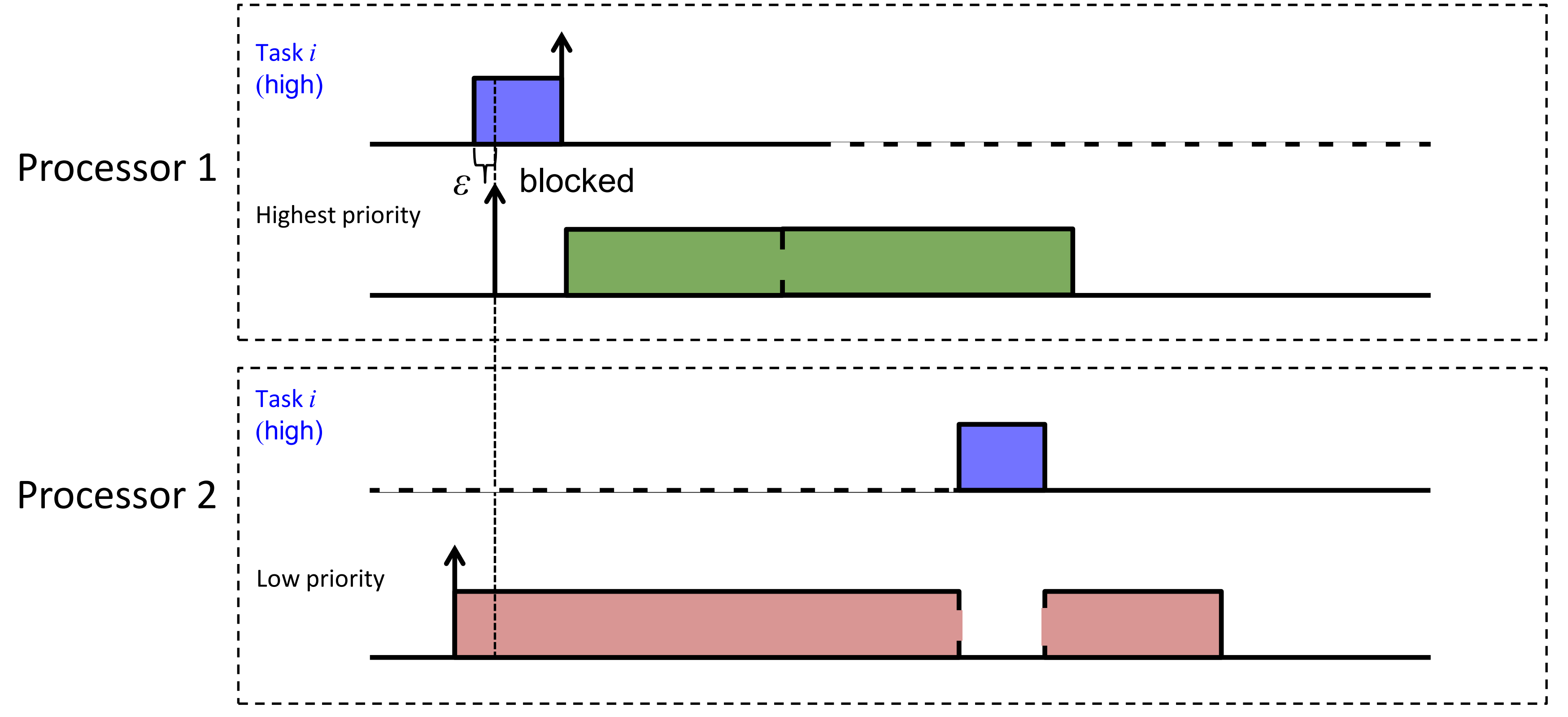
Schedulability Analysis under G-LP-FPS with Eager Preemptions



- Case 1: no "push through" blocking
- Case 2: presence of "push through" blocking

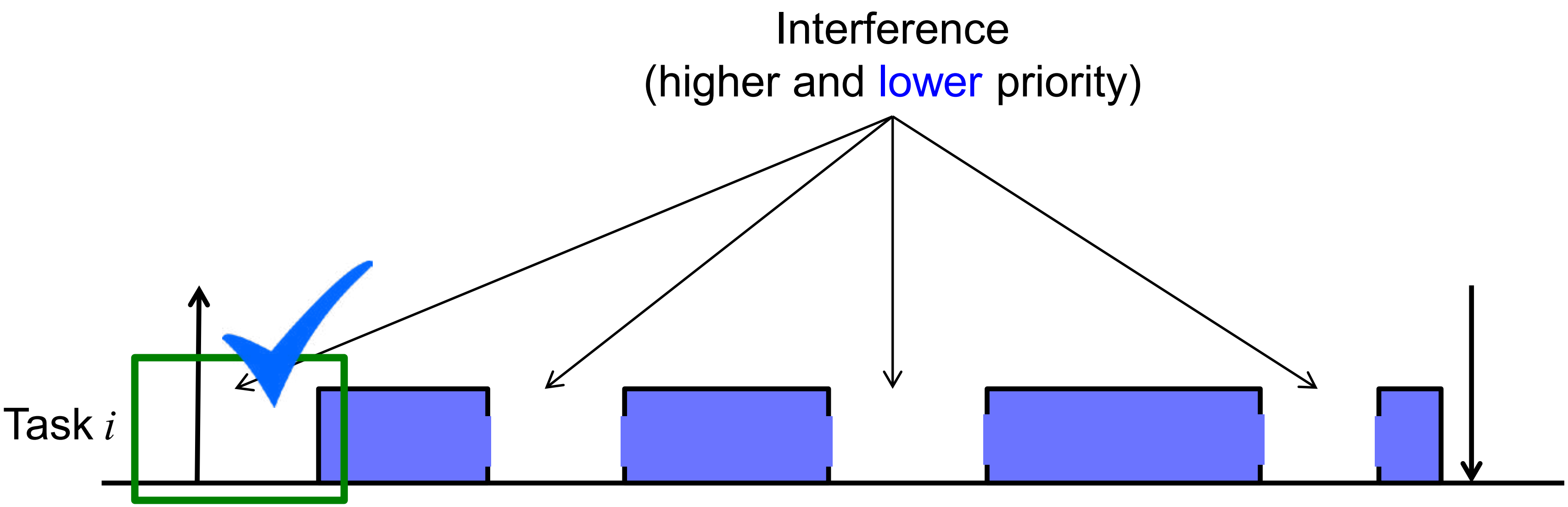
Lower Priority Interference before Task Start Time

Case 2: presence of push through blocking

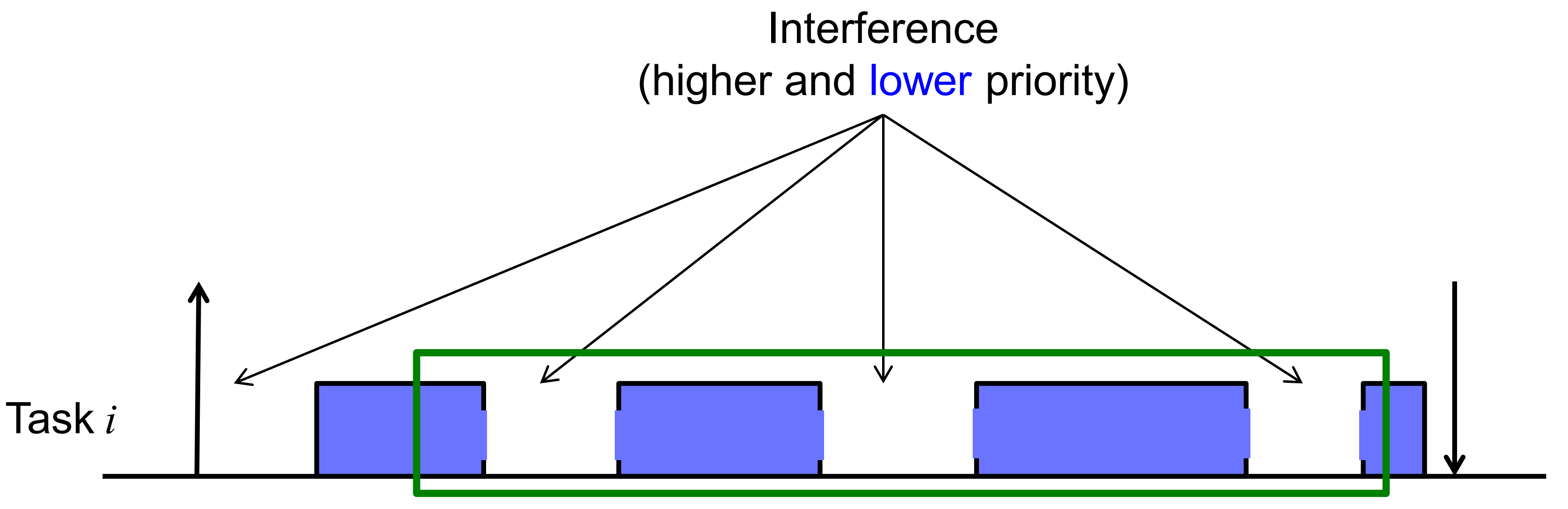


blocking= sum of m largest ({lower priority NPRs, final NPR of i})

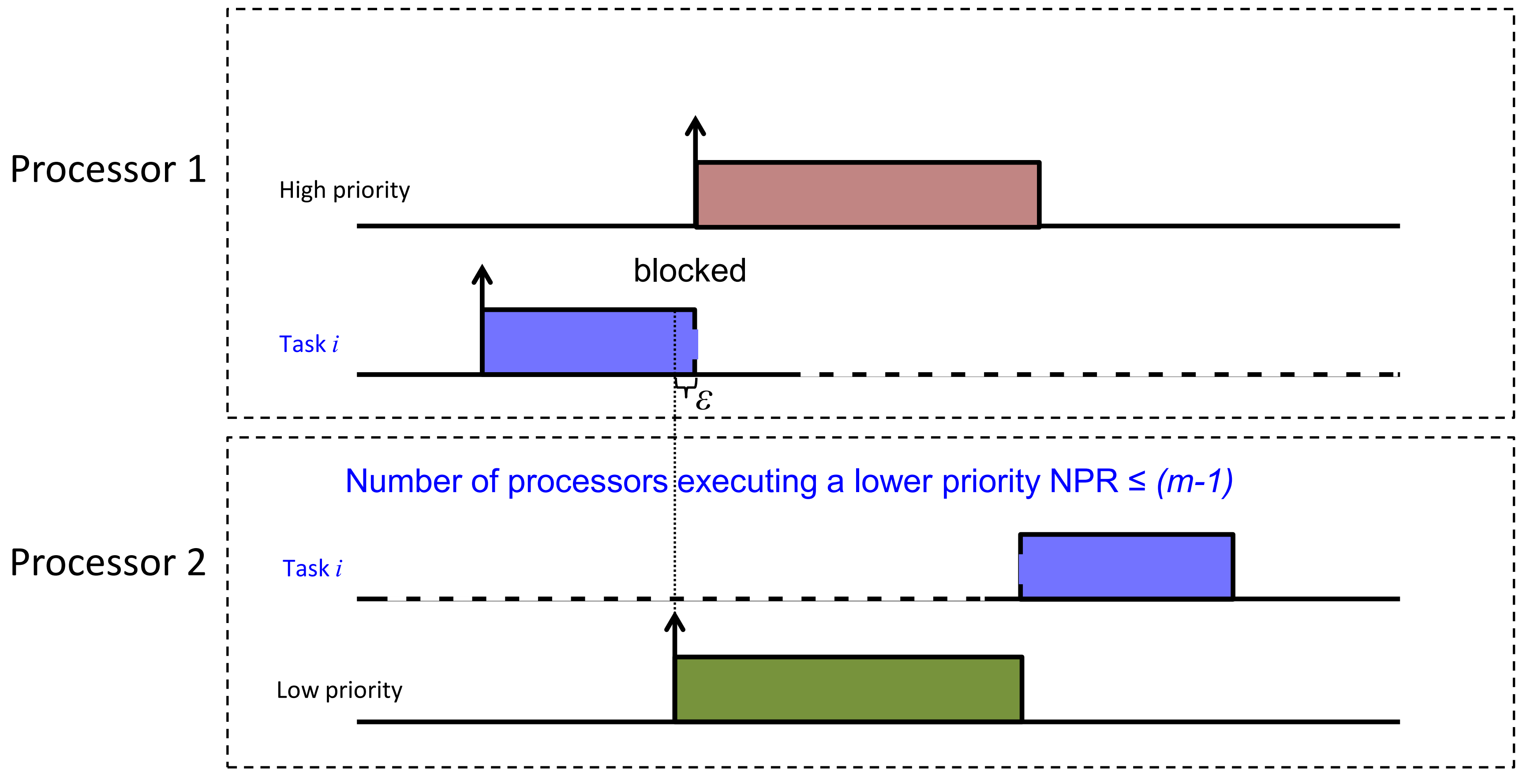
Schedulability Analysis under G-LP-FPS with Eager Preemptions



Schedulability Analysis under G-LP-FPS with Eager Preemptions

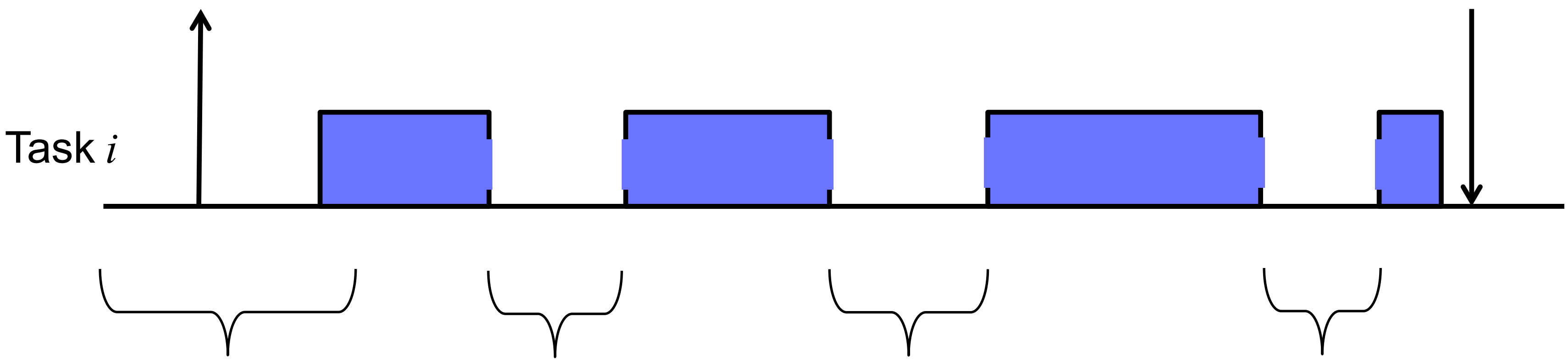


Lower Priority Interference after Task Start Time



blocking= sum of *(m-1)* largest ({lower priority NPRs})

Schedulability Analysis under G-LP-FPS with Eager Preemptions



$R_i =$ Interference (higher and lower priority) + Interference (higher and lower priority) + Interference (higher and lower priority) + Interference (higher and lower priority) $+ C_i$

Of course, preemption may not occur at all preemption points

- No. of preemptions as a function of response time to reduce pessimism
- Details in the paper

MÄLARDALEN UNIVERSITY SWEDEN · UNIVERSITY of York · BITS Pilani K K Birla Goa Campus · UNIMORE UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Experiments

Which among eager and lazy preemption approaches is better for Global Limited Preemptive FPS (G-LP-FPS)?

- Compared schedulability under eager preemptions and lazy preemptions
  - Test for lazy preemptions: test for link-based scheduling that implements lazy preemptions
    - Inflate task execution time with largest blocking time
    - Perform response time analysis for G-P-FPS

Overview of Experiments

- Task utilizations generated using UUnifastDiscard
- Periods in the range *50* to *500*
- Taskset utilization in the range *2.4* to *m*

- We investigated how *weighted schedulability* varies with:
  1. Varying number of tasks
  2. Varying number of processors
  3. Varying NPR lengths
     a. relatively large NPR *w.r.t* task WCETs
     b. relatively small NPR *w.r.t* task WCETs

Weighted Schedulability

- Weighs schedulability with utilization (Bastoni *et al.*, OSPERT'10)

$$W(p) = \frac{\sum_{\forall \Gamma} U(\Gamma) S(\Gamma, p)}{\sum_{\forall \Gamma} U(\Gamma)}$$

Weighted Schedulability

- Weighs schedulability with utilization (Bastoni *et al.*, OSPERT'10)

$$W(p) = \frac{\sum_{\forall \Gamma} U(\Gamma) S(\Gamma, p)}{\sum_{\forall \Gamma} U(\Gamma)}$$

Schedulability of taskset Γ $w.r.t$ parameter p

Weighted Schedulability

- Weighs schedulability with utilization (Bastoni *et al.*, OSPERT'10)

Utilization of taskset Γ

$$W(p) = \frac{\sum_{\forall \Gamma} U(\Gamma) S(\Gamma, p)}{\sum_{\forall \Gamma} U(\Gamma)}$$

Weighted Schedulability

- Weighs schedulability with utilization (Bastoni *et al.*, OSPERT'10)

$$W(p) = \frac{\sum_{\forall\Gamma} U(\Gamma) S(\Gamma, p)}{\sum_{\forall\Gamma} U(\Gamma)}$$

- Enables investigation of schedulability *w.r.t* a second parameter in addition to utilization

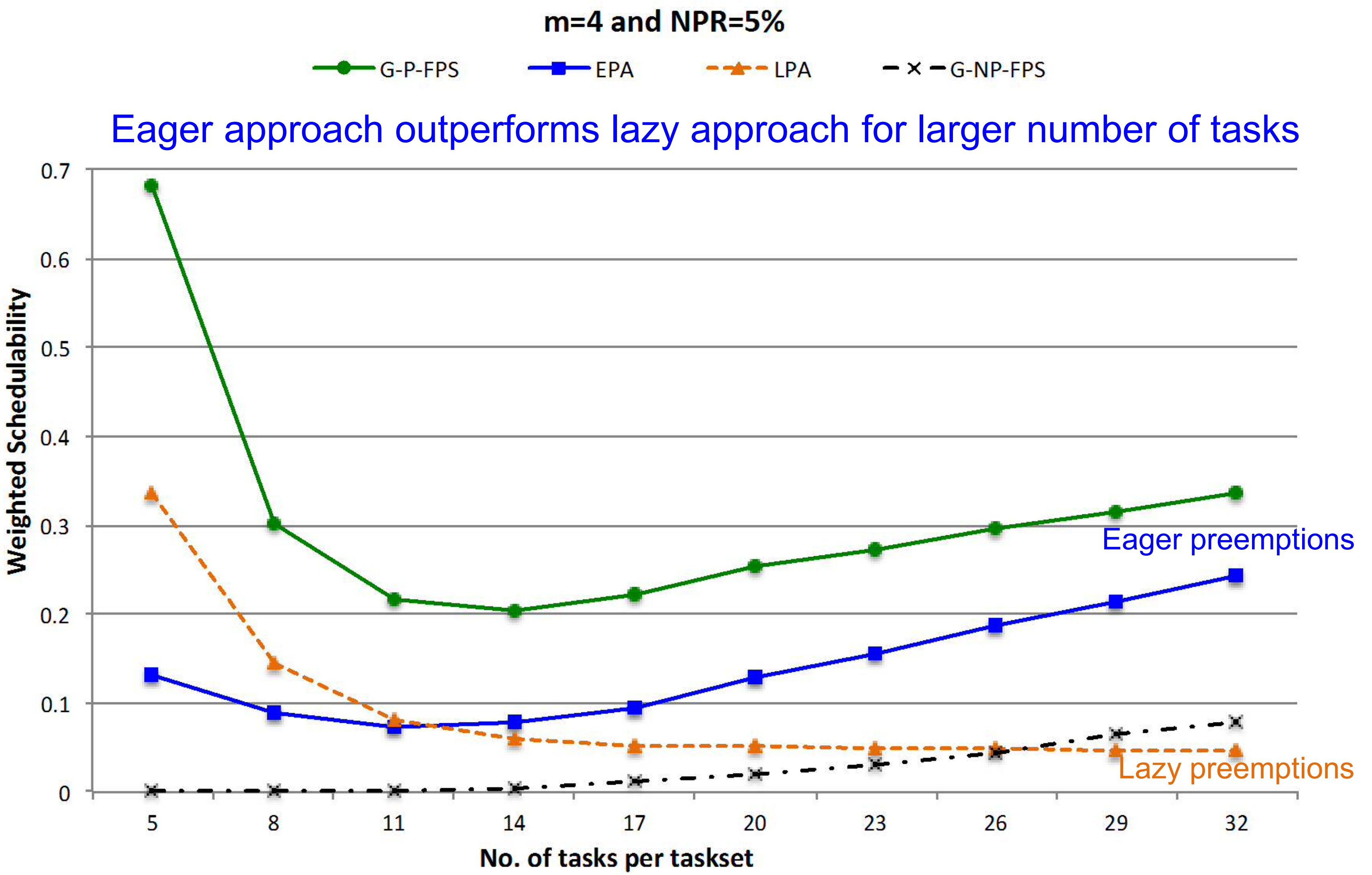
- Higher weighted schedulability implies a better algorithm with respect to scheduling high utilization tasksets (and thus better algorithm *w.r.t* efficiency)

Experiments

We investigated how *weighted schedulability* varies with:

1. Varying number of tasks
2. Varying number of processors
3. Varying NPR lengths
   a. relatively large NPR *w.r.t* task WCETs
   b. relatively small NPR *w.r.t* task WCETs

Varying Number of Tasks



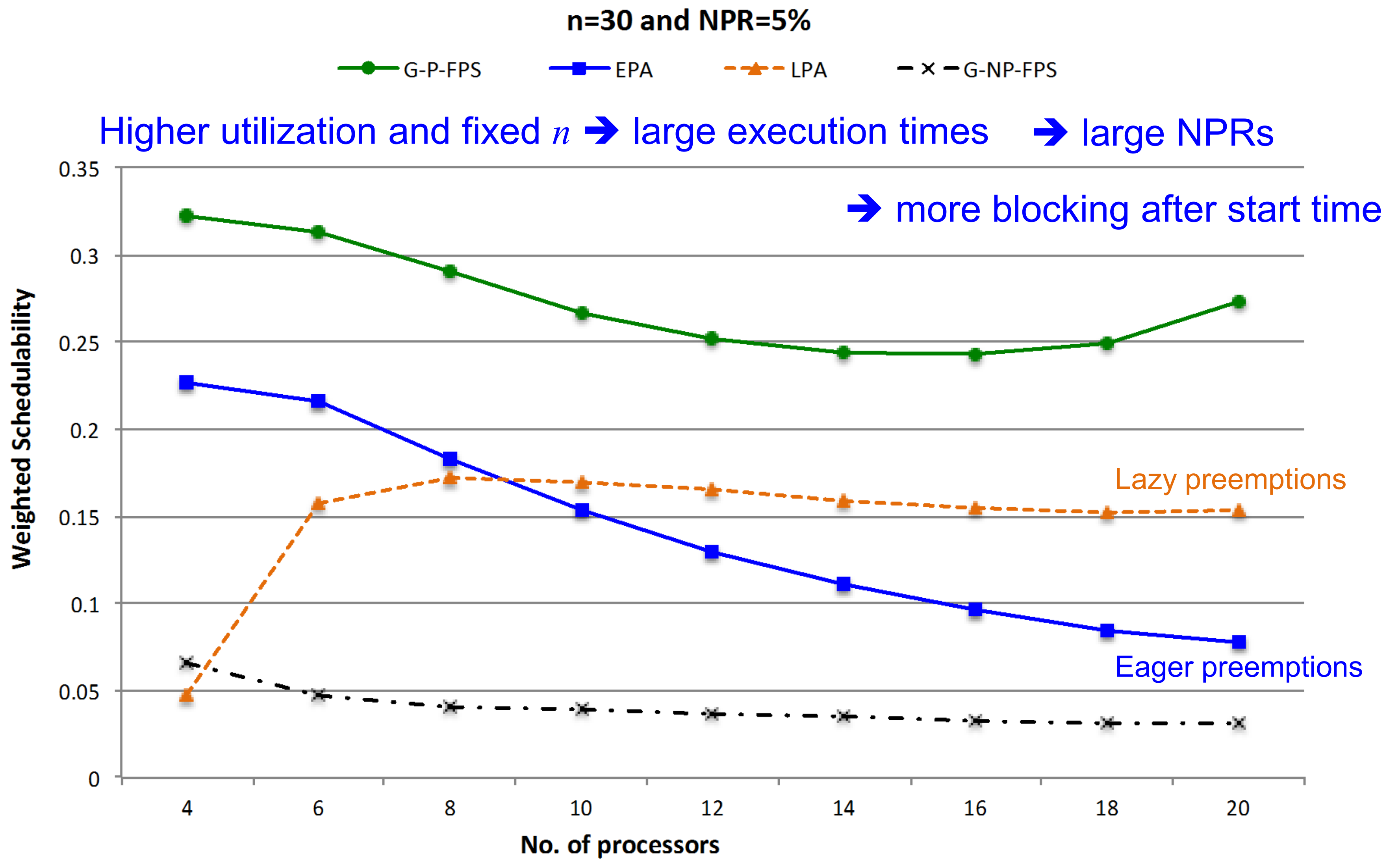
Experiments

We investigated how *weighted schedulability* varied with:

1. Varying number of tasks
2. Varying number of processors
3. Varying NPR lengths
    a. relatively large NPR *w.r.t* task WCETs
    b. relatively small NPR *w.r.t* task WCETs

MÄLARDALEN UNIVERSITY SWEDEN UNIVERSITY of York BITS Pilani K K Birla Goa Campus UNIMORE UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Varying Number of Processors



MÄLARDALEN UNIVERSITY SWEDEN

UNIVERSITY of York

BITS Pilani K K Birla Goa Campus

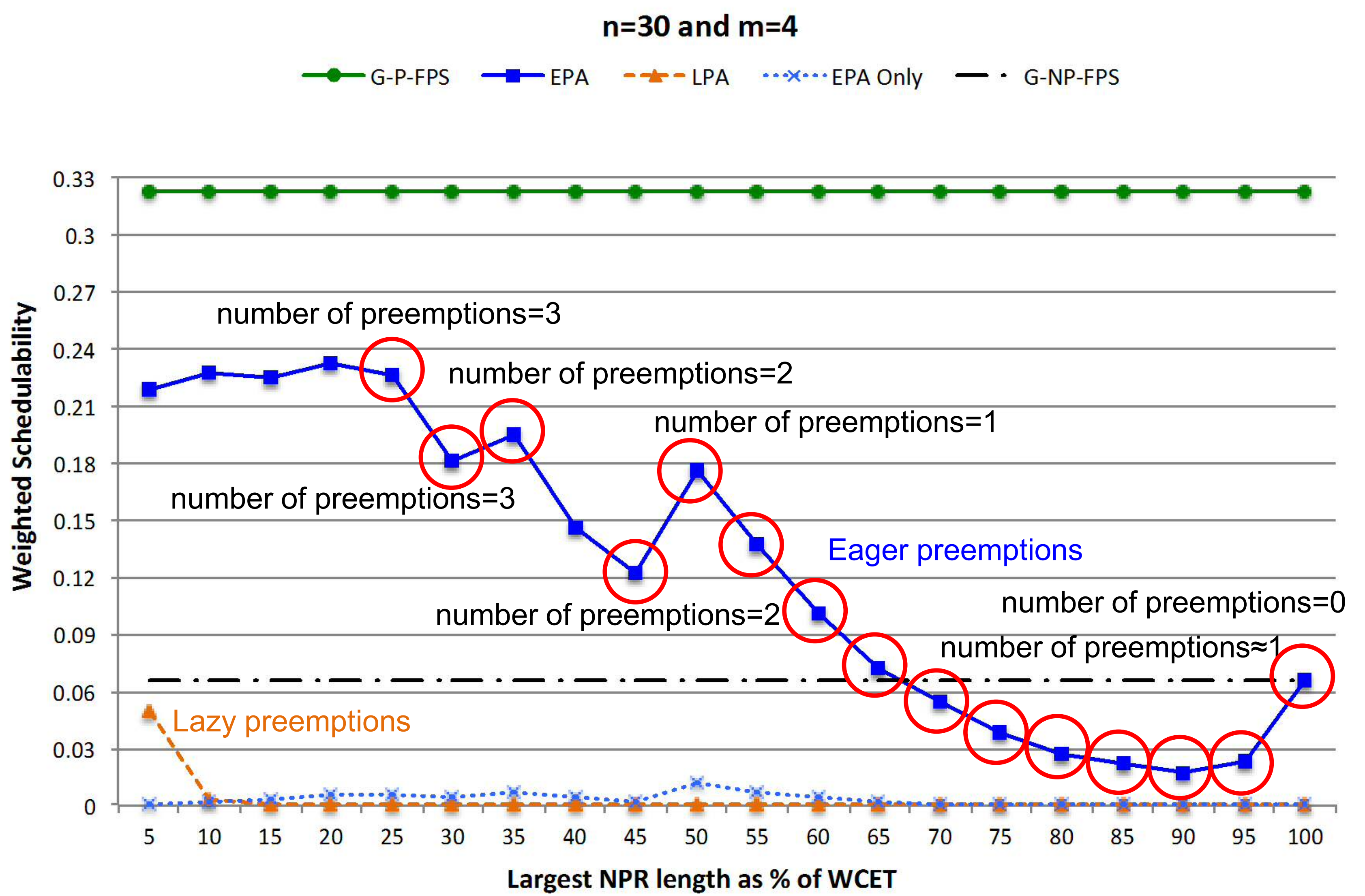
UNIMORE UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Experiments

We investigated how *weighted schedulability* varied with:

1. Varying number of tasks
2. Varying number of processors
3. Varying NPR lengths
    a. relatively large NPR *w.r.t* task WCETs
    b. relatively small NPR *w.r.t* task WCETs

Varying Lengths of NPRs (large)



MÄLARDALEN UNIVERSITY SWEDEN

UNIVERSITY of York

BITS Pilani K K Birla Goa Campus

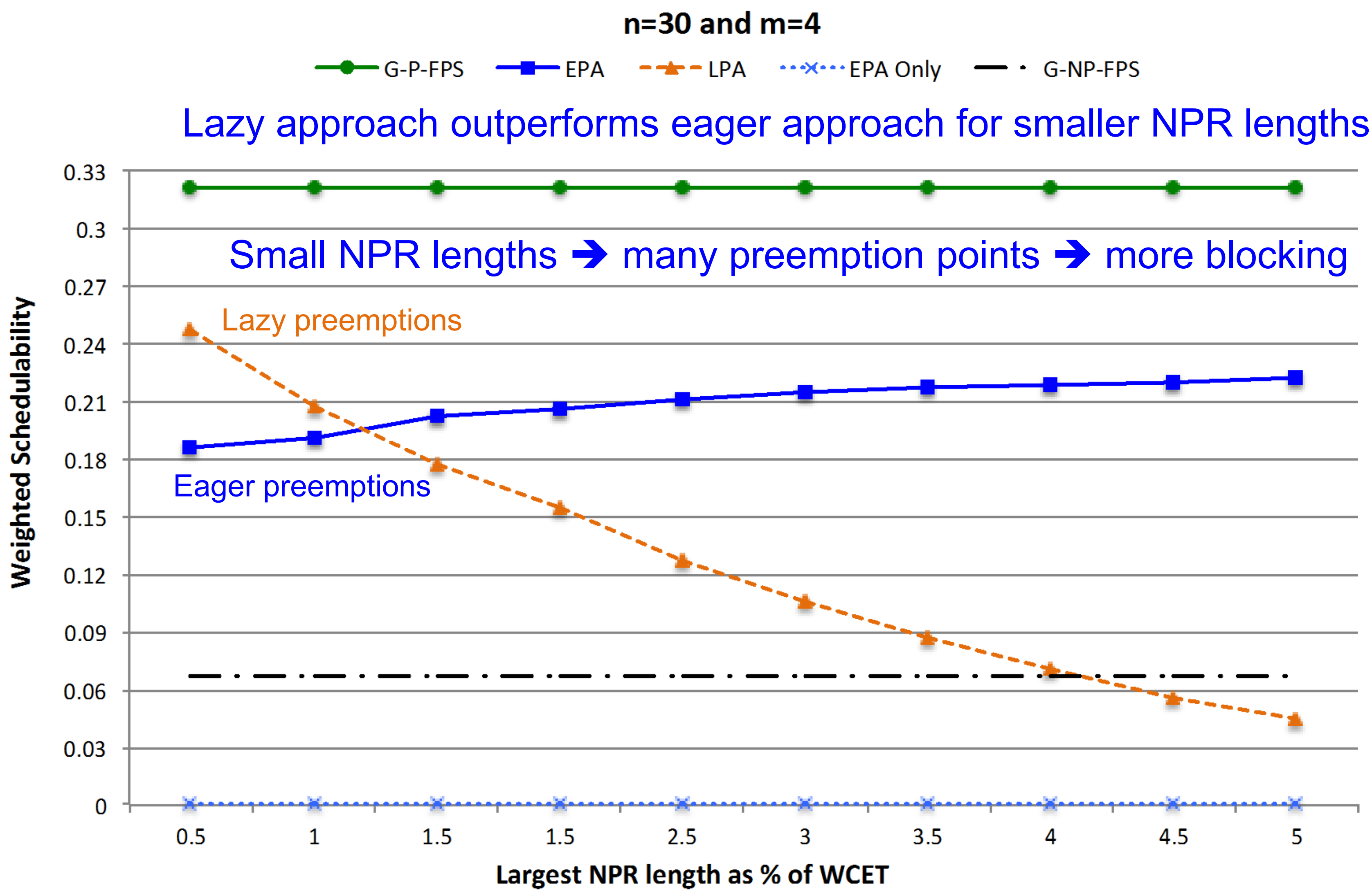
UNIMORE UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Experiments

We investigated how *weighted schedulability* varied with:

1. Varying number of tasks
2. Varying number of processors
3. Varying NPR lengths
   a. relatively large NPR *w.r.t* task WCETs
   b. relatively small NPR *w.r.t* task WCETs

Varying Lengths of NPRs (small)



MÄLARDALEN UNIVERSITY SWEDEN

UNIVERSITY of York

BITS Pilani K K Birla Goa Campus

UNIMORE UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Conclusions

- Presented a schedulability test for global LP FPS with eager preemptions
- Compared eager and lazy approaches using synthetically generated tasksets
  - Eager approach outperforms lazy approach
- Eager preemption is beneficial if high priority tasks have short deadlines relative to their WCETs
  - Need to schedule them ASAP
- Lazy preemption is beneficial if tasks have many preemptions points
  - Need to reduce blocking occurring after tasks start their execution

Future Work

- Evaluation of runtime preemptive behaviors of eager and lazy approaches under global EDF and FPS
  - LP scheduling with eager approach generates more runtime preemptions compared to preemptive scheduling (under submission to RTAS'16)
- Evaluation on a real hardware
  - Context Switch Overheads
  - Cache related preemptions delays
- Efficient preemption point placement strategies for multiprocessor systems

Thank you !

Questions ?

MÄLARDALEN UNIVERSITY SWEDEN

UNIVERSITY of York

BITS Pilani
K K Birla Goa Campus

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA